AD-764 124

BIOCYBERNETIC FACTORS IN HUMAN PERCEPTION AND MEMORY

STANFORD UNIVERSITY

PREPARED FOR
DEPARTMENT OF DEFENSE

MAY 1973

STANFORD UNIVERSITY

CENTER FOR SYSTEMS RESEARCH

ANNUAL REPORT

Biocybernetic Factors in Human Perception and Memory

Principal Investigator
David C. Lai

Information Systems Laboratory

SEL-73-026

ANNUAL REPORT

for the Advanced Research Projects Agency

of the Department of Defense

BIOCYBERNETIC FACTORS IN HUMAN PERCEPTION AND MEMORY

PRINCIPAL INVESTIGATOR:

Dr. David C. Lai

Contract Number: DAHC15-72-C-0232

ARPA Order Number: 2190

Technical Report Number: 6741-1

1 June 1972 to 31 May 1973

Department of Electrical Engineering
Stanford Electronics Laboratories
Stanford University Stanford, California 94305

PROFESSIONAL PERSONNEL

Dr. D. C. Lai, Visiting Professor of Electrical Engineering, Principal Investigator

Dr. T. Kailath, Professor of Electrical Engineering, Co-Principal Investigator

Dr. J. E. Anliker, Consultant

T. E. Attwood (B.S.E.E.), Scientific Programmer

K. H. Jacker (B.A. in Computer Science and B.A. in Mathematics), Scientific Programmer

M. Ein-Gal (M.S.E.E.), Graduate Student Research Assistant

A. Shah (M.S.E.E.), Graduate Student Research Assistant

M. Stauffer (B.S.E.E.), Graduate Student Research Assistant

FOREWORD

This Annual Report presents the accomplishments during the period of 1 June 1972 to 31 May 1973. The principal investigator is Dr. D. C. Lai and co-principal investigator is Dr. T. Kailath. Since the beginning of this project, it has been the result of the collaborative efforts of many individuals. Most of the staff members contributed in the writing of this report.

Biocybernetic Factors in Human Perception and Memory

Abstract

This project is concerned with the application of biocybernetic concepts to the problems of expanding human memory. ~~In particular,~~ the goal of ~~this~~ research is the enhancement of visual imagery as far as possible in the direction of photographic memory. Scanpaths of the eye during inspection of visual targets are treated as indicators of the brain's strategy for the intake of visual information. This research will determine the features that differentiate scanpaths associated with superior imagery from scanpaths associated with inferior imagery. Similarly, the features correlated with superior imagery ~~will be~~ differentiated from those correlated with inferior imagery. A computerized biocybernetic scheme will ~~be implemented in~~ an attempt to generate image enhancement and to train the individual to exert greater voluntary control over his own imagery. In this report, we give an account of our accomplishments in developing and implementing the necessary techniques for the realization of the biocybernetic scheme.

The implications of this project for the Department of Defense arise from the possibility of devising new and unusual techniques that will permit the development of stronger imagery in men with normal memories. Since it is definitely an asset for the military man to have the ability to absorb information rapidly and to retain it intact for long periods of time, this project will attempt to intensify the post-stimulus imagery as far as possible in the direction of photographic memory--and to search out those features of successful recall strategies so as to enhance the individual's power of redintegration.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

I. INTRODUCTION

This research project is concerned with the development of the biocybernetic concepts and techniques needed for the analysis and development of skills useful for the control and enhancement of human memory functions, particularly those related to the "eidetic" images. Specifically, we are concentrating on the problem of achieving biocybernetic expansion of human visual memory through the use of real-time computerized monitoring and feedback of cues that serve as keys to memory encoding and retrieval. This approach is based on the observation that the human nervous system depends heavily on spatial and temporal cues both in the encoding and retrieval of memories, especially sensory images.

Our plan is to develop and implement techniques for real-time monitoring and prediction of central nervous activities through the electroencephalographic signals and through the tracking of eye movements and eye fixations. We shall use these techniques to determine the temporal and spatial cues that serve as keys to human memory encoding and memory retrieval by tracking of brain states and eye positions. This information is then utilized for arranging the desired coincidences between various brain states, eye positions, and the delivery of visual stimulation. The visual stimuli are to be presented in complex tachistoscopic batteries or screens for either monopic or dichopic viewing. The eye movements and eye positions are to be measured by a unified system so as to obtain their optical and electrophysiological characteristics. The brain states are to be monitored through the EEG signals. Combining these real-time monitoring and

prediction techniques with a feedback scheme to close the control loop; we expect to obtain greater control of image persistence and image dissipation. In other words, we strive to use the computer system to supplement and strengthen those deficiencies in human memory that ordinarily result in image dissipation. We are assuming tentatively that a superior pattern of visual inspection that results in a superior memory is more consistent and also less probably of natural occurrence than a visual inspection strategy that is less consistent. Through the use of real-time computer guidance, we shall attempt to steer the subject toward improved encoding and decoding strategies for memory. In this report, we shall describe our accomplishments in designing, developing, and implementing the techniques for reaching these goals. The general schemes are described in the next section.

We chose a PDP-15 computer system manufactured by Digital Equipment Corporation (DEC) for its unique capabilities suited to real-time data acquisition and processing for the realization of our schemes and algorithms. A major effort was devoted to bringing up this new PDP-15 computer system to full operational status. This required staff familiarization with the computer as well as the design and implementation of assorted hardware, mainly for interfacing various laboratory instruments. The computer system is now being fully utilized for data acquisition, data analysis, and further software development. Several experimental programs and general graphics software have been completed. These are described in Section III.

A real-time monitoring technique for EEG signals is described in Section IV. Our work on the measurement and tracking of eye movement

is presented in Section V. Conclusion and future plans are made in Section VI.

The significance of this research project for the Department of Defense resides in the possibility of devising new and unusual techniques that will permit the training of stronger imagery in persons with normal memories. Since it is definitely an asset for the military person to have the ability to absorb information rapidly as well as to retain it intact for long periods of time, this project attempts to intensify the post-stimulus imagery as far as possible in the direction of photographic memory--and to identify and seek out those features for successful strategies of retrieval so as to enhance the redintegrative power of the individual.

II. DESCRIPTION OF GENERAL SCHEMES

Since this project is concerned with the development of the bio-cybernetic concepts and techniques necessary for the analysis and training of skills useful for the manipulation and control of the concrete ("eidetic") images, we plan to concentrate on those aspects of concrete memory which could be expected to make a substantial contribution to practical volitional memory if they can be brought under stronger individual control. It is imperative for us to develop for analyzing some of the basic strategies used in superior memorization and to compare them with strategies which result in an inferior mnemonic product. This would involve the use of advanced signal processing schemes, pattern recognition concepts, real-time monitoring and prediction, and advanced electronic devices to effect the control of stimulus and response variables in real time. Using these techniques, we expect to generate frequent and sustained combinations of stimulus-response relationships which have a low probability of natural occurrence. Our general scheme can be divided into three major areas:

(1) To develop and implement techniques for real-time monitoring of central nervous activities via the analysis of electroencephalographic signals and thus to determine the optimal times for the presentation of sensory stimuli.

(2) To develop and implement techniques for real-time monitoring of eye fixation and eye movement and thus to determine in the visual field the optimal location for sensory stimulation during impression and recall.

(3) To develop and implement techniques for feedback of the temporal and spatial information to produce the most effective stimulus-response relationships during impression and recall.

The first two basic areas necessitate the use of advanced signal processing and pattern recognition schemes. The last involves the use of cybernetics. All of them require real-time monitoring and prediction, and the use of advanced electronic systems. The following schemes are to be implemented.

Denote the sequence of parameters of interest measured or observed at time instants $t_1, t_2, \ldots, t_n$ by $Y_1, Y_2, \ldots, Y_n$ which are vectors. These vectors represent the parameters derived from EEG signals or eye-position or eye-movement measurement. For instance, in the case of eye-position measurement, we have

$$Y_n = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}_n \tag{II.1}$$

where $(y_1)_n$ and $(y_2)_n$ denote the horizontal and vertical coordinates of the eye position at the time t_n; in the case of EEG signals we have

$$Y_n = \begin{bmatrix} A_n \\ f_n \\ \Phi_n \end{bmatrix} \tag{II.2}$$

where A_n, f_n, and Φ_n signify the amplitude, frequency, and phase, respectively. Let $X_1, X_2, \ldots, X_{n-1}, X_n, \ldots$ be the sequence of the true parameters desired to be estimated or predicted. It then depends on the measurement instruments that Y_n may relate to X_n in the following ways:

$$Y_n = MX_n + N_n \tag{II.3}$$

$$Y_n = M_n X_n + N_n \tag{II.4}$$

$$Y_n = F[X_n, t_n] + N_n \tag{II.5}$$

where N_n representing measurement errors and noises introduced is random in nature. The first expression shows that Y_n related to X_n by a linear transformation as specified by the matrix M and an additive observation or measurement error term symbolized by N_n. The second expression signifies that Y_n is related to X_n by the time-varying linear transformation M_n plus an observation error N_n. The last one denotes that Y_n is related to X_n by a system of nonlinear transformations, plus the error term N_n. For example, if the observation devices are transducers, then the measurement is made directly; i.e., there will be no transformations and

$$Y_n = IX_n + N_n = X_n + N_n \tag{II.6}$$

with $M_n = I$, the identity matrix. In other words, the matrix M or $F[X_n, t_n]$ characterize the measurement instrument.

The desired parameter X is generated by some physiological system which may be characterized by differential or difference equations and can thus be expressed as

$$X_{n-\ell} = \Phi_{n-\ell,n} X_n \tag{II.7}$$

where $\Phi_{n-\ell, n}$ is the so-called transition matrix. In essence, the above expression says that since the parameter is generated by some physiological system the past, present, and future values can change only with certain structural constraints and are interrelated. The differential equation or equations are a mere abstraction of these structural

constraints which set the rules how the parameter values should vary with time.

Since any estimation or prediction is based on past observed data, it will be convenient to write the equation in such a way that the past observed data will be in the form as the input to the algorithms to be developed and implemented by digital computer. Considering the fixed-length nature of the computer memory, we shall introduce the total observation vector and error vector which represent the past k vectors up to the present vector:

$$Y_{(n)} = \begin{bmatrix} Y_n \\ \hline Y_{n-1} \\ \hline \vdots \\ \hline Y_{n-k} \end{bmatrix} \quad \text{and} \quad N_{(n)} = \begin{bmatrix} N_n \\ \hline N_{n-1} \\ \hline \vdots \\ \hline N_{n-k} \end{bmatrix} \tag{II.8}$$

Assuming that the measurement instrument is characterized by the matrix M and combining Equation (II.4), (II.7), and (II.8), we obtain:

$$\begin{bmatrix} Y_n \\ \hline Y_{n-1} \\ \hline \vdots \\ \hline Y_{n-k} \end{bmatrix} = \begin{bmatrix} M_n \\ \hline M_{n-1} \Phi_{n-1,\, n} \\ \hline \vdots \\ \hline M_{n-k} \Phi_{n-k,\, n} \end{bmatrix} X_n + \begin{bmatrix} N_n \\ \hline N_{n-1} \\ \hline \vdots \\ \hline N_{n-k} \end{bmatrix} . \tag{II.9}$$

Defining the matrix

$$L_n \triangleq \begin{bmatrix} M_n \\ \hline M_{n-1} \Phi_{n-1,\, n} \\ \hline \vdots \\ \hline M_{n-k} \Phi_{n-k,\, n} \end{bmatrix} \tag{II.10}$$

we then have

$$Y_n = L_n X_n + N_{(n)} \tag{II.11}$$

which relates the total observation vector to the present desired vector which represents the parameters of interest to be estimated or predicted. The matrix L_n encompasses all of the a priori decisions made for the filtering or prediction schemes, viz., how and when we observe and what we believe we are observing. The task now is to estimate X_n or $X_{n+\ell}$. In the former case, we would obtain an estimate of the present parameter values X_n based on the past k observed values Y_{n-k}, Y_{n-k+1}, . . . , Y_{n-1}, and the present observed values Y_n. This is called filtering or smoothing. In the latter case, we would obtain an estimate of the future parameter values $X_{n+\ell}$ at the ℓth future instants from the present based on the present and past observed values. This is the so-called "prediction". Both the estimated values of X_n and $X_{n+\ell}$ are to be updated whenever a new measurement Y_{n+1} is made. The criterion most popularly used for obtaining the best estimate is the so-called least square principle in which the mean square error between the desired values and the actual value estimated is minimized. Without going through the detailed derivations, we shall write the resulting best estimator in the least squares sense

$$\hat{X}_n = \left[(L_n^T L_n)^{-1} L_n^T \right] Y_{(n)} . \tag{II.12}$$

We shall not derive here the computational algorithms for the implementation of this estimator. There are some classical algorithms in existence. We are planning to devise our own algorithms to suit our needs. The least-square criterion is based on minimizing the summed (average) squared errors. In other words, the algorithm was set up in such a way as to minimize the sum of the square of the random

components obtained on each successive sample drawn from the observation process without any regard for the errors of the individual components. The weighted least-square criterion is used while each of the errors of the individual components is minimized. This leads to the following estimate

$$\hat{X}_n = \left[(L_n^T R_{(n)} L_n)^{-1} L_n^T R_{(n)}^{-1}\right] Y_{(n)} \tag{II.13}$$

where

$$R_{(n)} = E\left[N_{(n)} N_{(n)}^T\right] \tag{II.14}$$

which is the total covariance matrix of the input errors. The filtering idea discussed above may be summarized by the block diagram shown in Figure II.1.

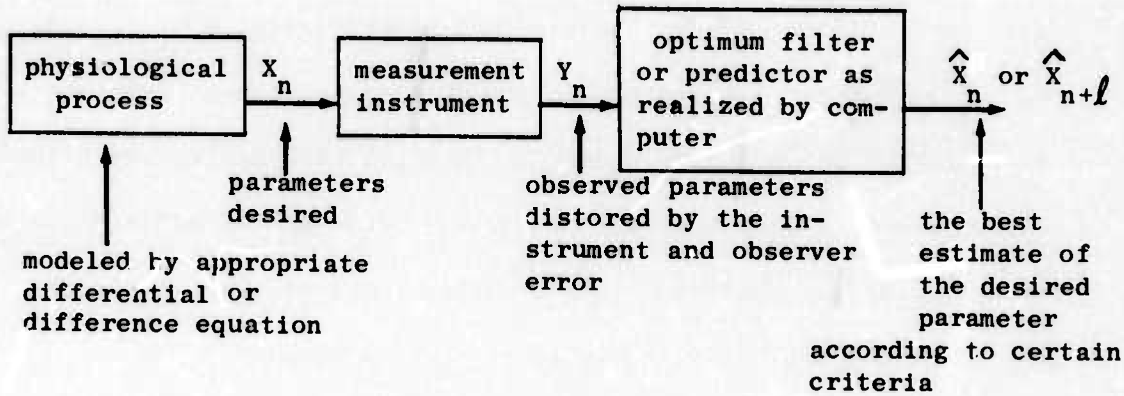


Figure II.1 Block Diagram Demonstrating the Basis for the Estimation or Prediction Scheme

To illustrate the use of the scheme described above, we shall consider the problem of measuring eye position and velocity of eye movement. To simplify the example, we shall assume that we are interested only in the motion along a single axis. The observation vector Y_n becomes a scalar y_n. Suppose these measurements were made every T

seconds (sampling interval). Let the most recent observed value be y_n. The total observation vector is then

$$Y_{(n)} = \begin{bmatrix} y_n \\ \hline y_{n-1} \\ \hline \vdots \\ \hline y_{n-k} \end{bmatrix} . \tag{II.15}$$

We shall use a polynomial $[\hat{p}(\gamma)]_n$ as our model. The variable γ is a time index increasing positively with the time whose origin is located at $t = (n-k)T$; i.e., kT seconds before most recent observation. Figure II.2 shows the γ-axis in relation with the time axis and the interpolating polynomial. When a new value y_n is observed after T seconds, the polynomial is updated from the new total observation vector

$$Y_{(n+1)} = \begin{bmatrix} y_{n+1} \\ \hline y_n \\ \hline \vdots \\ \hline y_{n-k+1} \end{bmatrix} \tag{II.16}$$

To be more definite, let the polynomial be of the first degree and K = 3, and we are interested in a one-step position and velocity predictor. Applying the least-square principle, but deleting the details, we obtain

$$\hat{x}_{n+1} = a_1 y_n + a_2 y_{n-1} + a_3 y_{n-2} + a_4 y_{n-3} \cdot \cdot \cdot \tag{II.17}$$

$$\hat{\dot{x}}_{n+1} = \frac{1}{T} (b_1 y_n + b_2 y_{n-1} + b_3 y_{n-2} + b_4 y_{n-3}) \tag{II.18}$$

where the values of the coefficients $a_1, \ldots, a_4$ and $b_1, \ldots, b_4$ can

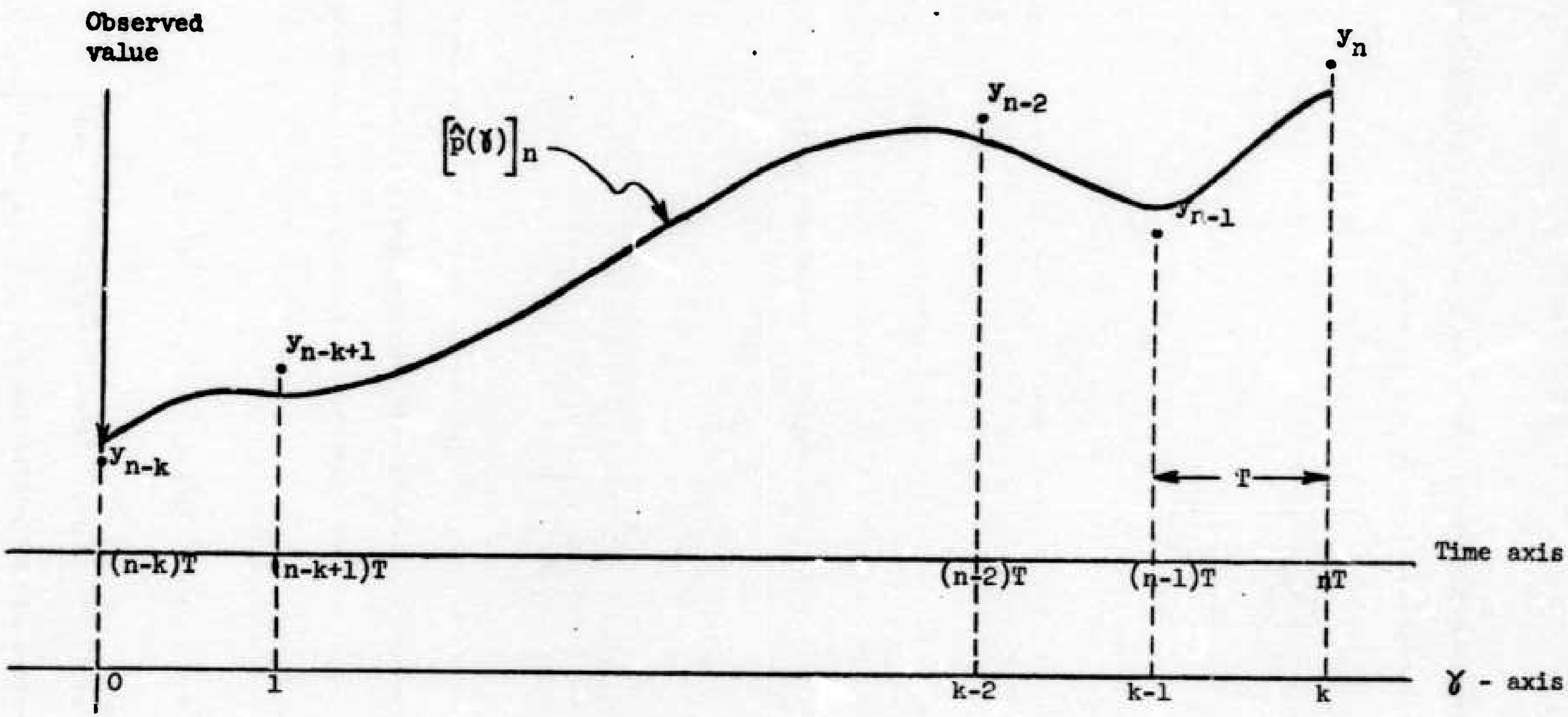


Fig. II.2 γ - Axis and Time Axis

be obtained by appropriately choosing the modeling polynomial and using the least-squares criterion. In the derivation, we have assumed the fixed-length nature of the filter memory. As each new observation y_n arrives, the previous ones are pushed down and y_{n-4} is discarded. The latest one-step predictions are computed by the use of (II.17) and (II.18). The predictor stores and uses only the most recent four observed values, and any which are older than 4T seconds are completely wiped out or forgotten. Figure II.3 shows the block diagram of our scheme.

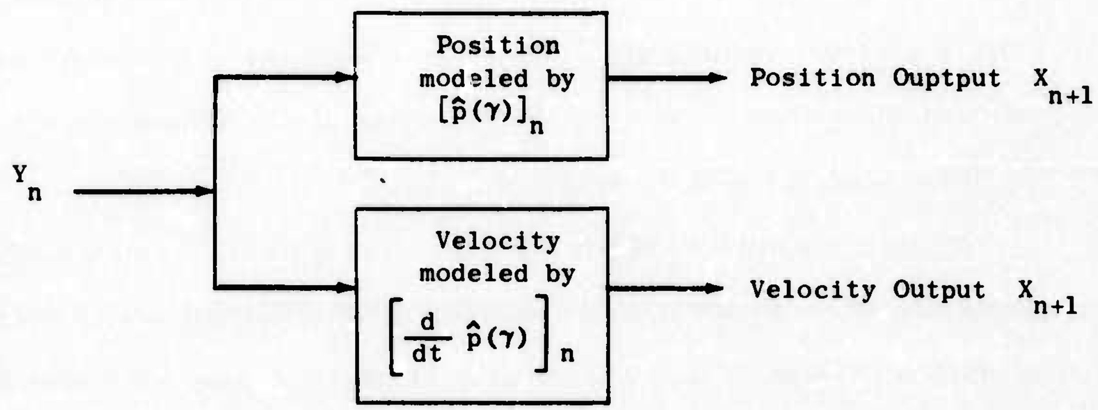


Figure II.3 Block Diagram Depicting the Prediction Algorithm

Such estimators must be computed simultaneously and independently for optical measurements and for electrooculographic measurements of eye position. These two sets of measurements must be compared frequently and combined into a single best estimator derived from a proper weighing of their respective contributions based on other factors, e.g., during eyelid closures, EOG information is obviously superior to optical information; on the other hand, when eyelids are open, the optical measurements can be used as a reference for nulling out EOG

drift or error. The scheme described above omits many details and refinements. It is intended solely to indicate the nature of the approach.

The success of the described scheme depends on a good model for the psycho-physiological process. Inappropriate modeling of the physiological process is a serious source of estimation error. The above-discussed scheme assumes that an appropriate model for the physiological process is known. The modeling problem, as we all know, is by itself a major research topic. This modeling problem may be circumvented by the use of a computer-based model with the ability to mimic the actual physiological system as far as the stimulus-response relationship is concerned. We shall discuss such a scheme as part of our integrated man-machine system.

Another assumption, which was made implicitly, is that the appropriate choice of parameters has been made. What parameters to measure is again a problem of major portions. Finally, we have also made the assumption that we can measure the selected relevant parameters separated from other irrelevant parameters or signals being generated by the physiological process. To achieve that, we plan to apply a variety of sophisticated filtering and signal processing techniques and to work in close collaboration with the life scientists who are skillful in making the various eye-pointing measurements.

As mentioned before, the success of the estimate or prediction of the psycho-physiological parameters hinges on an appropriate mathematical model. Since neither the characteristics of the physiological process

are completely known nor the characterization of the physiological process in mathematical terms is always possible. To circumvent this problem, we shall devise a computer-based model which is continually updated by a machine learning process. The model is to be realized by an algorithm which relates a set of stimuli, responses, and internal states. We will then use this computer-based model to predict (forward-time analysis) the sort of stimuli which should be used to the machine in order to produce the desired future model responses. In conjunction with this computer-based model, we plan to realize an optimal control scheme for delivery sequences of sensory stimulation conditionally related to eye position and brain state, and thus to explore systematically their relation to memory recall. To facilitate discussion we have simplified our integrated control scheme and depict it in Figure II.4. We use the EEG signal $(Y_n)_1$ and the eye-movement measurement $(Y_n)_2$ as our available outputs. These responses are closely controlled since the models mimic or are similar to the actual physiological process as far as the stimulus-response relation is concerned. The boxes in the block diagram of Figure II.4 are labeled according to their functions. All except the interface electronic system can be implemented on the PDP-15 computer.

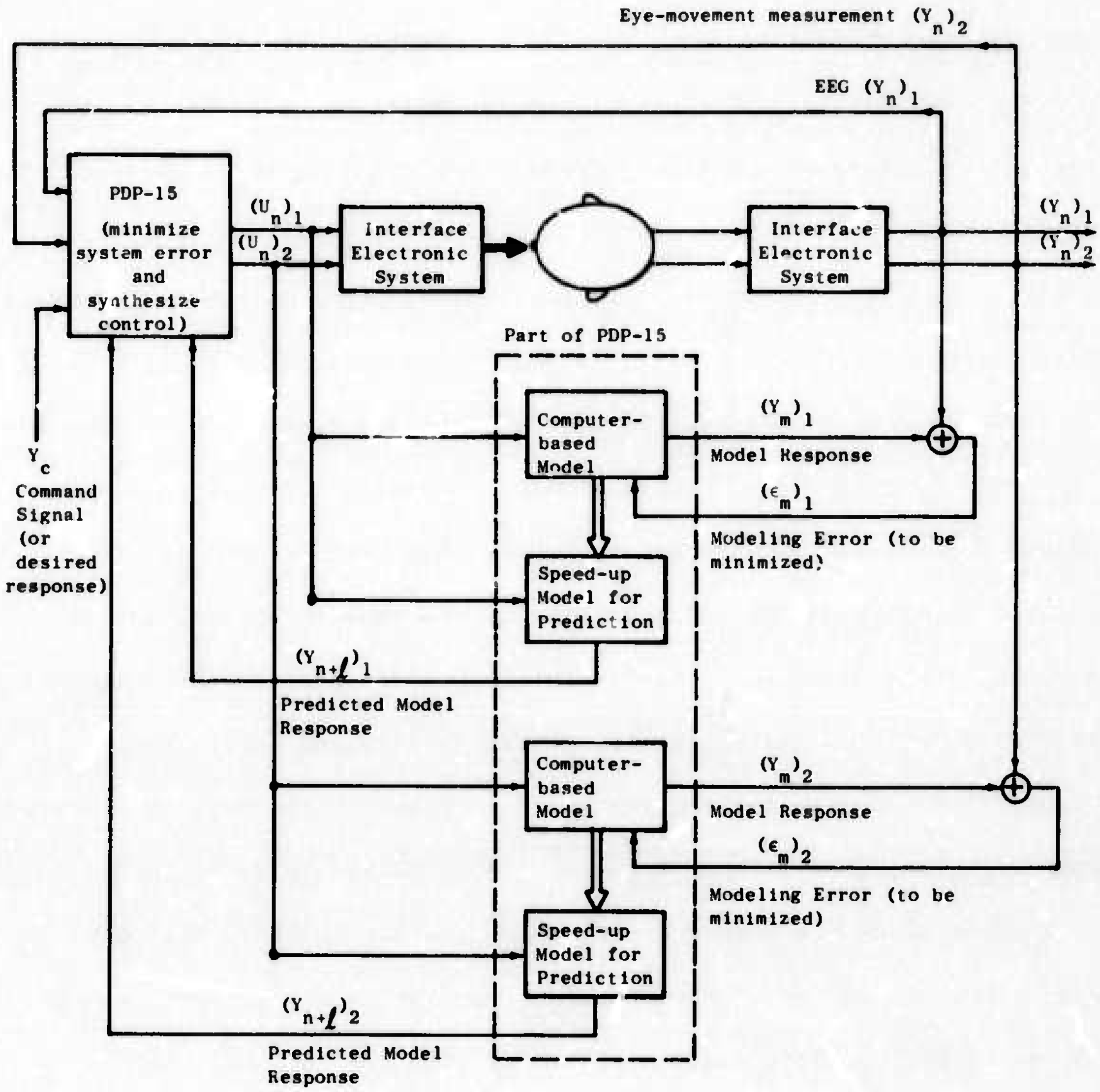


Figure II.4 A Control Scheme for Memory Tracking or Training

III. COMPUTER ORGANIZATION AND PROGRAMMING

The effort required to get an intermediate-sized computer facility running should not be underestimated. In this section, we describe our research facilities and our effort in designing interface hardware and programming in order to realize our schemes and algorithms. During the past year, one of our major efforts was devoted to bringing up the new PDP-15 computer system to full operational status. The computer system is now being fully utilized for both data acquisition and data analysis.

1. Hardware

The block diagram in Figure III.1 depicts our general set up. The PDP-15 computer system is the core of our hardware system which consists of the following items:

PDP-15/35 Disk Operating Advanced Monitor System, consisting of:
24,000 18-bit, 800-ns core memory
- LA30 DECWriter
- PC15 High Speed Paper Tape Reader/Punch
- KE15 Extended Arithmetic Element
- KA15 Automatic Priority Interrupt
- KW15 Real Time Clock
- TC15 DECTape Control
- TU56 Dual DECTAPE Transport
- RF15 DECdisk Control
- RS09 DECdisk Drive, 262,144 words

KM15 Memory Protect

KF15 Power Fail

FP15 Floating Point Processor

KT15 Memory Relocation

TC59D Magnetic Tape Transport Control for up to 8 TU10A, TU10B, TU30A, TU30B Magnetic Tape Transport Units

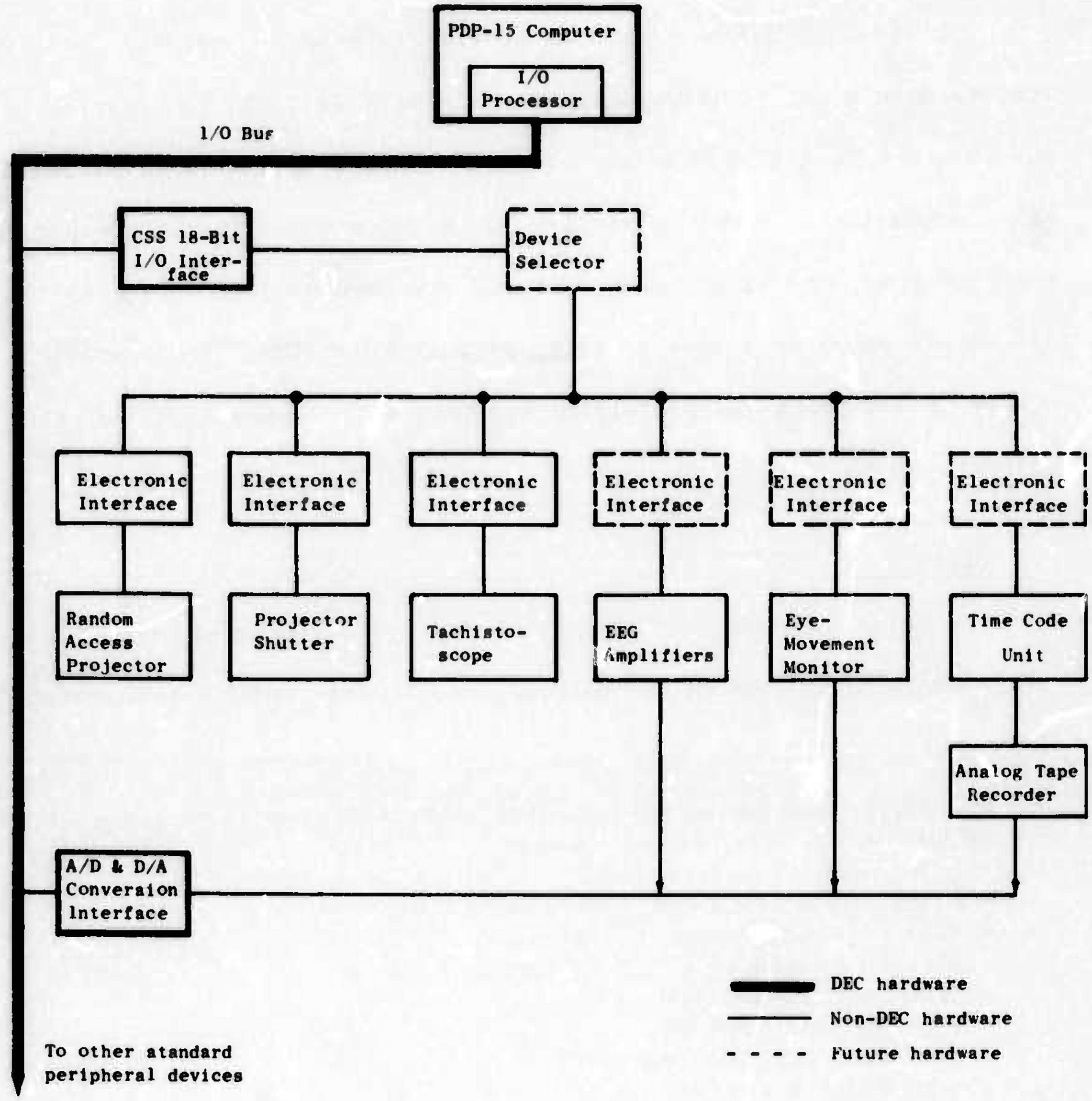


Figure III.1 Block Diagram of Hardware System

TU10A 9-Track, 45 ips Magnetic Tape Transport, 800 bpi

VP15C Oscilloscope Display, VR14 X-Y Display Unit (7" x 9" CRT) and Control

XY15AB 0.0005-Inch Step 18,000 Steps/Minute 31-inch drum, CalComp Model 563 and Control

CR15 300 cpm Reader and Control (Punched Cards)

AA15B Multiplexer Control for up to 16 Type AAC3 12-bit digital-to-analog converter channels

AAC3 Digital-to-Analog Converter, signal buffered, 0V to $\pm$ 10V

AD15 128 Channel A/D Converter (Medium Speed) Three-cycle data channel capacity with provision for mounting the first 32 channels; each 4-channel group requires one BA124

BA124 Four-Channel MOS FET Multiplex switch; one required for each 4-channel group

CSS 18-Bit Digital I/O Interface

RSX Software Package

RASP Software

LP15-F Line Printer

Figure III.2 shows the physical set up of the PDP-15 system and its peripherals. All of the interface hardware, shown in Figure III.1, except the CSS-18 I/O interface, are designed and built or to be built by us. A brief description of these interface hardware is given below.

(a) General

A general method for the interface of the special peripheral devices required for this project was developed using a special device (the CSS 18-bit interface) provided by the computer manufacturer. This is a bi-directional interface package which does all of the address decoding and timing functions required of a device controller. Our

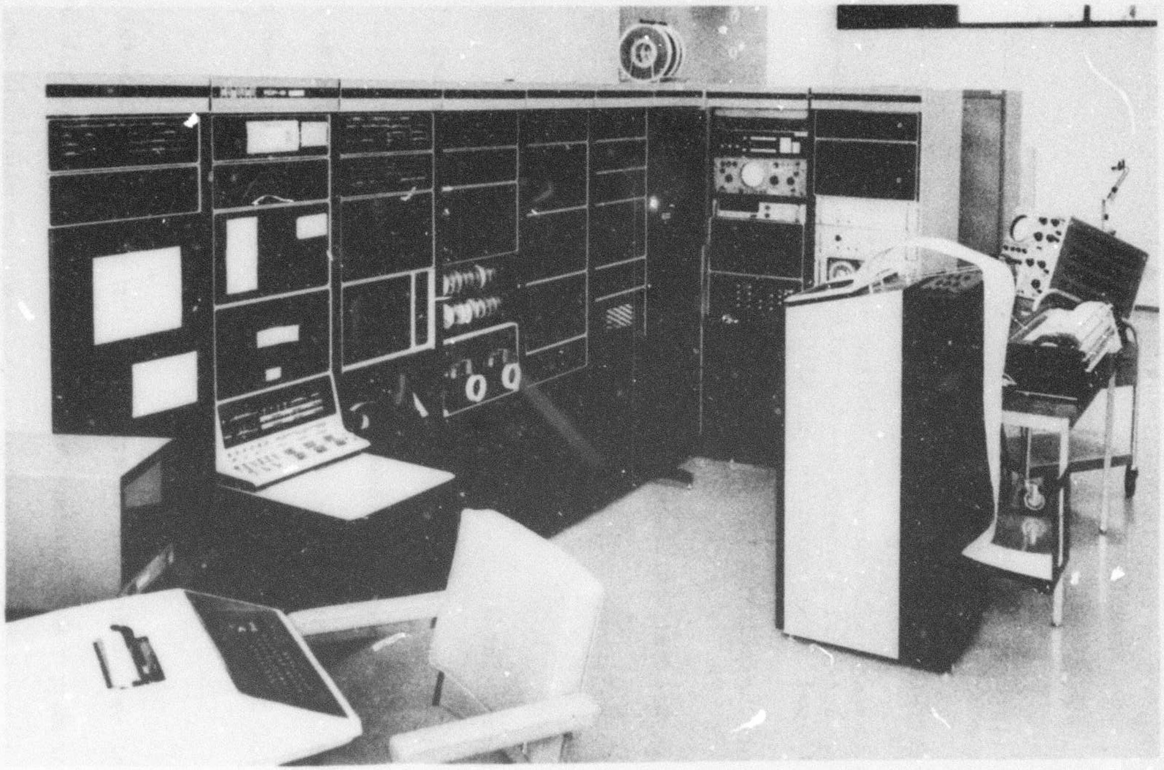


Figure III.2 The PDP-15 Computer and its Peripherals

use of several devices requires that a device selector/data multiplexer be built, and then additional special control (level shifting, inverting, etc.), unique to each device. Thus far, the special interface units have been designed, and, where noted, built. The device selector is in the planning stage. It is presently circumvented by paralleling several devices.

(b) Tachistoscope

The interface between the ICONIX tachistoscope and the PDP-15 has been completed and is operational. The unique functions required were level shifting and pulse generation, for a total of six commands.

(c) Random Access Slide Projector

The interface unit has been designed and is currently under construction. Logic was required to allow parallel entry of data, and to properly time the control signals. Level shifting of all command and data lines is also required.

(d) Time-Code Generator

The interface for the tape search unit/time code generator is not completed. The functions required are timing, control, and inversion of data.

2. Software

To bring up the PDP-15 computer to its operational stage and to fulfill our own special needs, we need to generate software which falls into the following three categories:

Category (a) General Support Programs and Subroutines

These include utility programs and library routines written for use by all project personnel. These are extensions of the operating system and utility packages produced by the manufacturer of the computer, which are required for day-to-day operation.

Category (b) Software Investigations

Research effort required for production of functional programs. Most work is concerned with the timing constraints on real-time system operation.

Category (c) Application Programs

These are programs which perform specific functions.

Most of the software in categories (a) and (b) is completed since these form the basis for operation of the system in our application. The software in category (c) is an on-going effort. We shall describe our accomplishment in each category separately.

(a) General Support Programs and Subroutines

The PDP-15 Real-Time Executive (RSX) was designed for a general scientific user. In order to tailor the system to our specific needs, certain modifications to the DEC software were necessary. Unfortunately, these modifications could not be realized using the RSX system (due mainly to the large core requirements of the resident monitor). We were, therefore, forced to use DEC's disk-oriented batch-processing monitor (DOS) to update and assemble the RSX system.

The first such modification was extending the RSX teletype handler

to simulate horizontal tabulation on our high-speed, teletype-compatible teleprinter. Unlike a teletype, the LA30 has no hardware tabulation. Additional core was added within RSX to monitor the current position of the carriage, make note of new read/write requests, and print blank characters when necessary to achieve carriage movement corresponding to tabulation. We are now able to 1) enter columnar data with much less mental effort, and 2) reduce symbolic file space by at least a factor of two.

In order to allow the system to be "booted in" from disk without using the paper tape reader, we made a second extension to RSX. The system must be "booted in" to start the system in the morning and to refresh memory following any severe system failure. The delivered system required a "warm start" tape to be placed in the paper tape reader along with the setting of certain console switches. By placing a copy of the warm start program within the resident monitor (less than 20 words) we are now able to warm start the system by simply depressing the START switch. Since paper tape is cumbersome to work with, this saves time and effort.

In addition to a powerful operating system, most computer users need an assortment of utility routines such as memory and magnetic tape dumps, graphics support (plotter, CRT), special purpose subroutines, magnetic tape copy, etc. Although the DOS system supplies some utility software, RSX includes virtually none. A list of the utility software written by us is given below. The details of this software effort are described in the Appendices A, B, and C.

Appendix A: GS-15 Graphics System

Appendix B: DIGIT Digitizing System

Appendix C: Miscellaneous Utility Software

- CORE/DISK DUMP
- DECtape Dump
- Magnetic Tape Dump
- LAS/ISWTCH
- RIPOFF
- REA
- GT132
- EMMXY
- CALDT/CALTD
- CALPLT
- CALCHK

(b) Software Investigation

Practically all of our raw data are in analog form. It is imperative that we should have, in operation, a good analog-to-digital converter used to input data. An I/O handler package, produced by the manufacturer, was available, and several programs were written to take advantage of this handler. A measurement method was designed to utilize a square wave input (at 1GHZ) and an off-line data analysis program which searches through the data to determine the consistency and accuracy of the A/D operation. It became obvious that accuracy of the A/D was not a significant cause of error, but that millisecond level scheduling of the data collection task was causing severe CPU utilization problems. The strategy chosen was to tune both the hardware and the software for optimum A/D operation.

Accordingly, the standard I/O handler was replaced with a high speed, less general driver which reduced the need to schedule two tasks virtually concurrently. In addition, a hardware modification was designed to allow the A/D to operate on its own clock rather than

requiring a CPU driven task to schedule each A/D operation. In combination, these will allow a substantial incrase in the digitizing rate that can be supported by the system.

(c) Applications Programs

These programs perform specific functions as specified by users. So far, we have developed and completed programs designed for use in acquisition of the eye-movement data. These can also be used for the acquisition of EEG data.

Another program (SEER) is in the development stage. This program is designed to use the Tektronix 611 CRT for image production and viewing. Both of these programs are described in detail in Appendix D.

IV. REAL-TIME MONITORING OF EEG SIGNALS

1. Introduction

Stroboscopic stimulation of the eyes results in changes of the alpha activity of the EEG. When the waveform is treated as a band-limited signal it is common to specify it by parameters like amplitude, phase, and frequency. Measurement and interpretation of these parameters (or any other set of parameters) during stimulation may supply information relevant to improvement of visual task performance.

Although spatial distribution of the electric field and particularly spatial coherence might be of even greater interest because of large integrating area of scalp electrodes, we have confined our attention to spatially-averaged temporal coherence which appears to be a dominant factor in a real-time measurement, interpretation, and prediction of the above-mentioned parameters.

In Section 2, we present an analytic treatment of the temporal coherence problem that leads to the basic scheme which we use in order to get rid of the redundancy associated with the α-waveform; this redundancy is, incidentally, an obstacle to the design of a real-time digital system. Section 3 presents the basic scheme and the digital processing associated with a representation of α-waves in a more compact way. In Section 4, we present the modified scheme.

2. Averaging with Coincident Peaks

A procedure commonly used to evaluate parameters associated with α-waveform is "averaging with coincident peaks". This method involves averaging a large number of "sample-functions" of recorded data which

are previously aligned so as to have a local maximum at the origin. The averaged signal looks like a damped sinusoid, the envelope of which decreases more or less exponentially [1], [2]. The rate of decrease serves as a measure of the temporal coherence, and models have been proposed to investigate the relationship between the rate of decrease and responsiveness to visual input.

We show that under certain conditions the averaged signal is a good approximation to the autocorrelation function of the in-phase and quadrature components of the analytic signal associated with the α-waveform. We conclude, therefore, that the familiar representation of the waveform by two slowly varying processes; viz., the in-phase and quadrature modulators, is applicable to the α-waveform although the latter, strictly speaking, does not fall into the category of narrow-band processes.

The following is a procedure for analyzing the envelope of a waveform obtained by averaging N sample functions of band-pass process conditioned to have a local maximum at t = 0, that is, averaging with coincident peaks.

Let $x_i(t), i = 1, \ldots N$, denote identically distributed independent band-pass processes defined by

$$x_i(t) = c_i(t) \cos \omega_0 t + s_i(t) \sin \omega_0 t \tag{IV.1}$$

where $c_i(t)$, $s_i(t)$, $c_j(t)$, $s_j(t)$ $i, j, = 1 \ldots N$ are stationary normal processes with zero mean and

$$E[c_i(t)c_j(t-\tau)] = E[s_i(t)s_j(t-\tau)] = r(\tau)\delta_{ij} \tag{IV.2}$$

$$E[s_i(t)c_j(t)] = 0 \; \forall \, i, j .$$

It follows that the autocorrelation of $x_i(t)$ is

$$r_x(\tau) = r(\tau)\cos\omega_0\tau\ . \tag{IV.3}$$

Now define the coincident-peaks-sum process $x(t)$

$$x(t) = \frac{1}{N}\sum_{i=1}^{N} x_i(t-t_i) \tag{IV.4}$$

where t_i is an instant of local maximum and is given by

$$\tan\omega_0 t_i = \left.\frac{c_i(t) + \omega_0 s_i(t)}{\omega_0 c_i(t) - s_i(t)}\right|_{t = t_i} \tag{IV.5}$$

if the band-width is narrow with respect to ω_0 we may omit $c_i(t)$ and $s_i(t)$ and get

$$\tan\omega_0 t_i \approx \frac{s_i(t_i)}{c_i(t_i)} = \mathrm{tg}\phi_i \tag{IV.6}$$

$$\cos\omega_0 t_i \approx \frac{c_i(t_i)}{\epsilon_i(t_i)};\ \sin\omega_0 t_i \approx \frac{s_i(t_i)}{\epsilon_i(t_i)}$$

where the phase $\phi_i(t)$ and the envelope $\epsilon_i(t)$ are defined by

$$\tan\phi_i(t) = \frac{s_i(t)}{c_i(t)} \tag{IV.7}$$

$$\epsilon_i^2(t) = s_i^2(t) + c_i^2(t) \tag{IV.8}$$

we conclude that

$$x_i(t_i) = \epsilon_i(t_i)\cos(\omega_0 t_i - \phi_i) = \epsilon_i(t_i) \tag{IV.9}$$

which means that shifting the time origin by t_i is equivalent to the initial conditions:

$$s_i(0) = s_{i0} \geq 0; \; c_i(0) = c_{i0} \geq 0;$$

$$s_{i0}^2 + c_{i0}^2 = \epsilon_i^2(0) = \epsilon_{i0}^2 \quad . \tag{IV.10}$$

Now we write x(t) in the form

$$x(t) = \frac{1}{N} \sum_{i=1}^{N} x_i(t) = \frac{1}{N} \sum_{i=1}^{N} c_i(t) \cos\omega_0 t + \frac{1}{N} \sum_{i=1}^{N} s_i(t) \sin \omega_0 t$$

$$= c(t) \cos\omega_0 t + s(t) \sin \omega_0 t = \epsilon(t) \cos(\omega_o t - \phi(t)) \tag{IV.11}$$

where

$$c(t) = \frac{1}{N} \sum_{i=1}^{N} c_i(t); \; s(t) = \frac{1}{N} \sum_{i=1}^{N} s_i(t)$$

$$\text{and } \tan\phi(t) = \frac{s(t)}{c(t)}; \; \epsilon^2(t) = s^2(t) + c^2(t) \; . \tag{IV.12a-d}$$

The conditional mean and variance of the random variables $c_i(t)$ and $s_i(t)$ given the initial conditions are

$$E[s_i(t) \mid s_{i0}, c_{i0}] = r(t) s_{i0}$$

$$E[c_i(t) \mid s_{i0}, c_{i0}] = r(t) c_{i0} \tag{IV.13a-c}$$

$$\text{Var}[s_i(t) \mid s_{i0}, c_{i0}] = \text{Var}[c_i(t) \mid s_{i0}, c_{i0}] = r_0[1-\rho^2(t)]$$

where $\rho(\tau)$ is the normalized autocorrelation

$$\rho(\tau) = \frac{r(\tau)}{r(0)} \; ; \tag{IV.14}$$

Defining the N-dimensional vectors $\underline{c_0}$, $\underline{s_0}$ by their respective components c_{i0}, s_{i0}, we have due to independence

$$E[c(t) \mid \underline{c_0}, \underline{s_0}] = r(t) \cdot \frac{1}{N} \sum_{i=1}^{N} c_{i0} = r(t)c(0) = r(t)c_0$$

$$E[s(t) \mid \underline{c_0}, \underline{s_0}] = r(t) \cdot \frac{1}{N} \sum_{i=1}^{N} s_{i0} = r(t)s(0) = r(t)s_0$$

(IV.15a-c)

$$\sigma^2(t) = \mathrm{Var}[c(t) \mid \underline{c_0}, \underline{s_0}] = \mathrm{Var}[s(t) \mid \underline{c_0}, \underline{s_0}] = \frac{1}{N} r_0 [1-\rho^2(t)]$$

Defining $\mu(t)$ by

$$\mu^2(t) = E^2[s(t) \mid \underline{s_0}, \underline{c_0}] + E^2[c(t) \mid \underline{c_0}, \underline{s_0}] = r^2(t)\, \epsilon_0^2 \qquad \text{(IV.16)}$$

we can write the conditional distribution and the conditional mean of the envelope $\epsilon(t)$ as

$$\rho(\epsilon \mid \underline{s_0}, \underline{c_0}) = \frac{\epsilon}{\sigma^2} \,\mathrm{Exp} \left[- \frac{\left(\epsilon^2 + \mu^2\right)}{2\sigma^2} \right] I_0 \left[\frac{\epsilon\mu}{\sigma^2} \right] \qquad \text{(IV.17)}$$

$$E[\epsilon \mid \underline{s_0}, \underline{c_0}] = \sqrt{\frac{\pi}{2}}\, \sigma \,\mathrm{Exp}\, (-\alpha)[(1+2\alpha) I_0(\alpha) + 2\alpha I_1(\alpha)] \qquad \text{(IV.18)}$$

where $I_0(\cdot)$ and $I_1(\cdot)$ are the modified Bessel function of order zero and one, respectively, and α is given by

$$\alpha = \frac{\mu^2}{4\sigma^2} = \frac{r_0 N \epsilon_0^2}{4} \; \frac{\rho^2(t)}{1-\rho^2(t)} \; . \qquad \text{(IV.19)}$$

Assuming that $\lim_{t\to\infty} r(t) = 0$, we get the Rayleigh distribution for the envelope whose mean is decreasing as $N^{-1/2}$:

$$\lim_{t\to\infty} \rho[\epsilon \mid \underline{s_0}, \underline{c_0}] = \frac{N\epsilon}{r_0} \;\mathrm{Exp} \left[\frac{-N\epsilon^2}{2r_0} \right] \qquad \text{(IV.17a)}$$

$$\lim_{t\to\infty} E[\epsilon \mid \underline{s_0}, \underline{c_0}] = \sqrt{\frac{\pi}{2} \frac{r_0}{N}} \quad . \tag{IV.18a}$$

These expressions do not depend on $\underline{s_0}$, $\underline{c_0}$ so averaging over all proper values of $\underline{s_0}$, $\underline{c_0}$ will not affect them. Keeping in mind that $x_i(t)$ has a local maximum at the origin, we can write

$$\lim_{t\to\infty} \rho(\epsilon) = \frac{N\epsilon}{r_0} \text{ Exp}\left[- \frac{N\epsilon^2}{2r_0}\right] \quad , \tag{IV.17b}$$

$$\lim_{t\to\infty} E(\epsilon) = \left(\frac{\pi r_0}{2N}\right)^{1/2} \quad . \tag{IV.18b}$$

It would be nice to average out ϵ_0 from formulas (IV.17) and (IV.18); however, ϵ_0 is a sum of N independent random variables with identical Rayleigh distribution.

We could not obtain a closed expression for the distribution of ϵ_0, but for the large N it could be approximated by a Gaussian distribution. Noting that $N\to\infty$ and/or $t \to 0$ implies $\alpha\to\infty$, and using asymptotic approximations to the Bessel functions

$$I_n(x) \sim (2\pi x)^{-1/2} \text{ Exp}[x] \qquad x >> 1 \ , \tag{IV.20}$$

we get for the conditional distribution of the envelope

$$\rho(\epsilon \mid \underline{c_0}, \underline{s_0}) \approx \frac{\epsilon}{\mu}^{1/2} (2\pi\sigma^2)^{-1/2} \text{ Exp}\left[\frac{-(\epsilon-\mu)^2}{2\sigma^2}\right] \approx (2\pi\sigma^2)^{-1/2} \text{ Exp}\left[\frac{-(\epsilon-\mu)^2}{2\sigma^2}\right] . \tag{IV.17c}$$

The conditional mean is obtained by expanding (IV.18) in powers of α^{-1}

$$E[\epsilon \mid \underline{c_0}, \underline{s_0}] \approx \mu (1 + \frac{1}{8\alpha}) = \epsilon_0(1 + \frac{1}{8\alpha}) \cdot \rho(t) \tag{IV.18c}$$

which shows that for small t and/or large N the conditional mean of the envelope is proportional to the autocorrelation. In order to average out ϵ_0 we note that for large N the distributuion density $\rho(\epsilon_0)$ is concentrated about the mean:

$$E(\epsilon_0) = \left(\frac{\pi r_0}{2}\right)^{1/2} .$$

In that case, we approximate the average over ϵ_0 by inserting $E(\epsilon_0)$ for ϵ_0 and get

$$p[\epsilon(t)] \approx (2\pi\sigma^2)^{-1/2} \ \mathrm{Exp}\left[\frac{-[\epsilon-\rho E(\epsilon_0)]^2}{2\sigma^2}\right] , \qquad \text{(IV.17d)}$$

$$E[\epsilon(t)] \approx \left(\frac{\pi r_0}{2}\right)^{1/2} \left(1 + \frac{1}{\pi N} \cdot \frac{\rho^2}{1-\rho^2}\right) \cdot \rho(t) \approx \left(\frac{\pi r_0}{2}\right)^{1/2} \rho(t) \qquad \text{(IV.18d)}$$

for large N and/or small t.

Note that unless $\rho(t) = 0$ the envelope $\epsilon(t)$ tends to $\rho(t)$ if N is sufficiently large. For large t, $\rho(t)$ tends to zero and the envelope tends to its constant asymptote which is N-dependent

$$\lim_{t\to\infty} E[\epsilon(t)] = \left(\frac{\pi r_0}{2N}\right)^{1/2} .$$

By taking N to be very large, the envelope converges to its mean which is proportional to the autocorrelation r(t).

The evaluation of the temporal coherence is reduced to the problem of estimating the auto and cross-correlation functions of the two slowly varying in-phase and quadrature modulators. Moreover, it is more efficient (from the system viewpoint) to deal with the slowly varying signals, provided the representation of the α-waveform by means of the two slowly varying signals is accurate. This observation led

us to the basic scheme presented in the next section.

3. The Basic Scheme: Representation and Monitoring

We describe here a new technique for real-time monitoring and predicting brain states via EEG signals. The EEG waveform is assumed to be a narrow-band process whose spectrum is centered around a mean frequency which may be slowly varying. Designating the EEG signal by $f(t)$, we have

$$f(t) = e(t)\cos[\theta(t)] = e(t)\cos[\omega(t) + \varphi(t)]$$
$$= s(t)\sin\omega_o t + c(t)\cos\omega_o t \quad .$$

By tracking the center frequency ω_o, we may represent the signal by the two processes $s(t)$ and $c(t)$. If the process is stationary whose spectrum is symmetric about ω_o, then $s(t)$ and $c(t)$ are uncorrelated so the prediction would be easier to deal with.

To obtain the in-phase component $c(t)$ and the quadrature component $s(t)$, we use a discrete "phase-locked loop" system described in the block diagram shown in Figure IV.1. This system tracks the center frequency ω_o and resolves the EEG waveform into $c(t)$ and $s(t)$. Once $c(t)$ and $s(t)$ are obtained for a window width of the EEG signal, we reduce the data by taking the mean value, the mean slope, etc. This is done by fitting a linear segment to the data in the window through recursive filtering. The values thus obtained serve as data vector components.

The key to the performance of this real-time monitoring technique is the discrete phase-locked loop system depicted in Figure IV.1. This

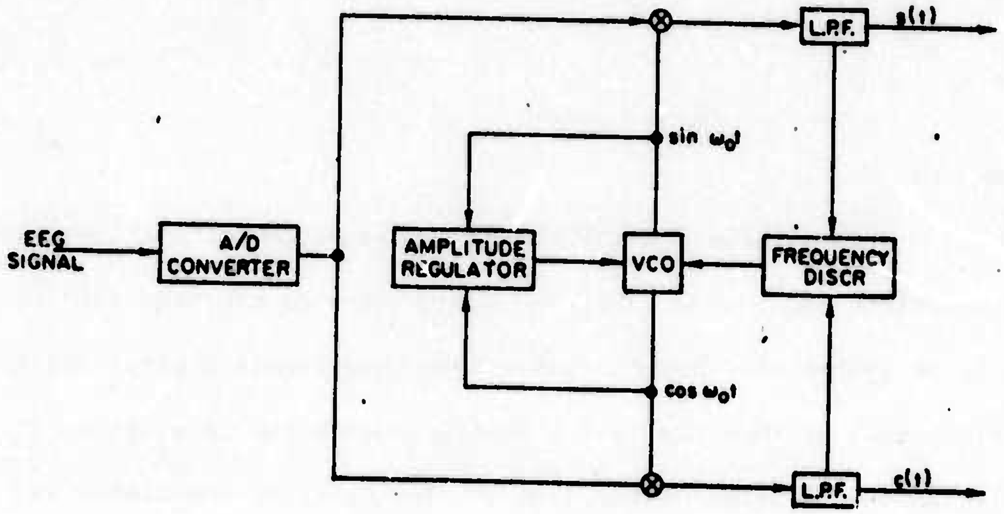


Figure IV.1 Resolver of
In-phase and Quadrature Signals

system includes:

(a) Voltage-Controlled Oscillator

The VCO is realized by generating the sine and cosine wave forms through the difference equations

$$S(N) = \alpha S(N-1) + \beta C(N-1)$$

and $$C(N) = \alpha C(N-1) - \beta S(N-1) .$$

Since the frequency and the amplitude are functions of α and β, we control these parameters in such a way as to maintain a constant amplitude and to change the frequency in accordance with the output of the discriminator. Specifically, we maintain unit amplitude by making $\alpha^2 + \beta^2 \simeq 1$ and increase or decrease the frequency by increasing or decreasing the parameter β according to the output of the discriminator.

(b) Frequency Discriminator

The frequency discriminator changes the frequency information in s(t) and c(t) to voltage which is linearly proportional to the

frequency.

(c) Low-Pass Filters

The low-pass filters are a standard 3-pole Butterworth filter with a 3-db bandwidth about 5 Hz. The two output signals c(t) and s(t) of the previous system are then fed separately into linear digital filters that fit to each of them the best linear approximation in a window of k samples in the least mean square sense. The width of the window was chosen to be 0.15 sec. The two parameters of the linear approximation, viz., the mean and the slope, are calculated recursively by digital filters known as the "comb" and "slope" filters. These filters are the digital realization of the transfer functions given by

$$C(jx) = \frac{\sin Kx}{\sin x}$$

$$S(jx) = \frac{\sin[(K+1)x]}{4\sin^2 x} - \frac{(K+1)\cos Kx}{4\sin x}$$

where C(jx) and S(jx) are the amplitude response of the comb and slope filters, respectively; and x is the normalized frequency. The value for K is 63 which corresponds approximately to the window width of 0.15 sec. The normalized frequency x is calculated by

$$x = \frac{\omega T}{2}$$

where T is the sampling period (2.5 msec), ω is measured in radians/second.

The phase characteristics of the two filters are of the same linear type

$$\text{phase}(\omega) = \frac{(K-1)}{2}\,\omega T\ .$$

These filters are block diagrammed in Figure IV.2.

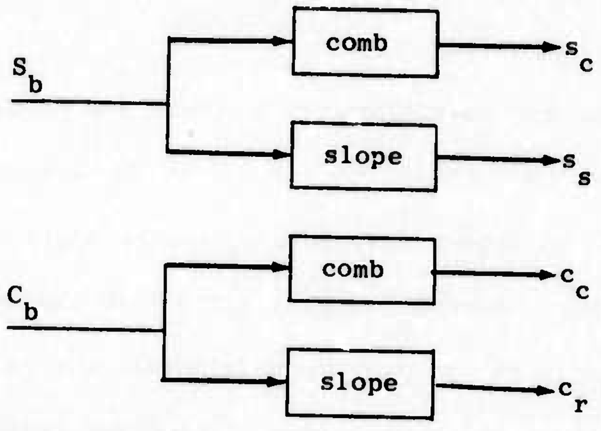


Figure IV.2 Block Diagram for Linear Fitting Filters

It should be noted that the four outputs of these filters contain information about the frequency and may be used as a frequency discriminator. If higher order "derivatives" are needed, they can be obtained by using recursive filters in a similar fashion.

4. The Modified System

The design of the modified system was motivated by the desire to get an accurate representation of the α-waveform by two slowly varying signals without any phase delay or distortion.

We designed and implemented a digital real-time system for obtaining the representation of a particular EEG rhythm in its in-phase and quadrature components with the constraint that the error is zero when synthesized. In a way, this system provides a form of data compression. The resultant representation can be advantageously employed for data transmission and data storage as well as real-time monitoring of either

psycho-physiological states or pathological states. For these applications, the requirement of an error-free synthesis or reproduction of the EEG signal is imperative.

A common scheme for obtaining the in-phase and quadrature components of a signal is to multiply the signal by an oscillator output at quadrature with the center frequency f_o of the signal and then passing the resultants through low-pass filters in the forward path. However, this method does not provide a faithful replica of the signal when the two components are synthesized. The error arises mainly from the phase distortions of the filter or the pure delays of the non-recursive filter. Our digital system uses, instead, an oscillator running at twice the center frequency of the signal and a filter in the feedback path. This filter does not affect the reproduction fidelity; however, it plays a role in determining certain statistical properties of the slowly varying components of the resultant representatives. Hence, one may choose a filter to minimize certain properties of the outcome such as the bandwidth, variance, etc.

Let the input signal in its digitized form be y(N). We shall define a two-dimensional oscillator vector H(N) by

$$H^T(N) \triangleq \left[\cos 2\pi f_o N \quad \sin 2\pi f_o N\right] . \qquad \text{(IV.21)}$$

We may write

$$y(N) = X^T(N)H(N) \qquad \text{(IV.22)}$$

where X(N) is a two-dimensional vector whose components $x_1(N)$ and $x_2(N)$ are the slowly-varying components sought after. Note that the

Equation (IV.22) is not unique since there exist many X(N) which could satisfy (IV.22). For a certain particular choice, $x_1(N)$ and $x_2(N)$ are related by Hilbert transform. Let $\hat{X}(N|N-1)$ be a function of the past values of X(N). For instance, $\hat{X}(N|N-1)$ could be a linear combination of X(0), X(1), ..., X(N-1). Denote the difference between X(N) and its $\hat{X}(N|N-1)$ by

$$E(N) = X(N) - X(N|N-1) \ . \tag{IV.23}$$

The norm of the difference is

$$\|E(N)\| = \left[E^T(N)E(N)\right]^{1/2} \ . \tag{IV.24}$$

We choose X(N) by minimizing (IV.24) but satisfying (IV.22) simultaneously. The solution thus obtained is

$$X(N) = \tfrac{1}{2}\left[I - A(N)\right]\hat{X}(N|N-1) + H(N)y(N) \tag{IV.25}$$

where I is the identity matrix and A(N) is the matrix with its components given by the double-frequency oscillator; i.e.,

$$A(N) = \begin{bmatrix} \cos 4\pi f_o N & \sin 4\pi f_o N \\ \sin 4\pi f_o N & -\cos 4\pi f_o N \end{bmatrix} \ . \tag{IV.26}$$

If $\hat{X}(N|N-1)$ is the best linear estimator of X(N) based on its past history, then the representation of the input signal y(N) by the vector E(N) corresponds to the innovation representation $\nu(N)$ with

$$\overline{\|E(N)\|^2} = \overline{\nu^2(N)} \tag{IV.27}$$

where $\nu(N) = y(N) - \hat{y}(N|N-1)$ and $\hat{y}(N|N-1)$ is the best linear estimator

of y(N) in terms of its past values y(0), y(1), ..., y(N-1).

The digital system is depicted by a block diagram shown in Figure IV.3. The oscillator is realized by a second order difference equation whose coefficients are controlled by the amplitude regulator and the output of the frequency discriminator. We have selected an EEG signal* for analysis by our digital system. In Figure IV.4 we show both the original EEG signal* and the EEG signal synthesized from these two components. Note the high fidelity of the reproduction. The filter in the feedback path was chosen to minimize the first order difference $\| X(N) - X(N-1) \|$.

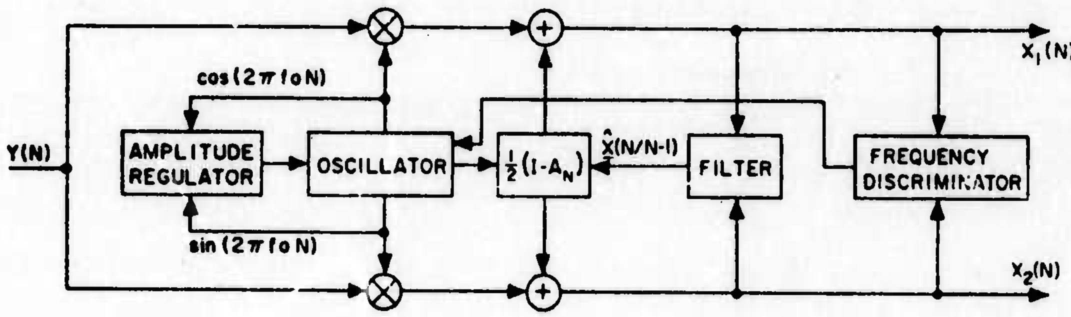


Figure IV.3 Block Diagram of the Digital System for Resolving the Signal into In-phase and Quadrature Components

*The EEG data were furnished by Dr. J. E. Anliker of NASA-Ames Research Center.

1.

2.

3.

4.

5.

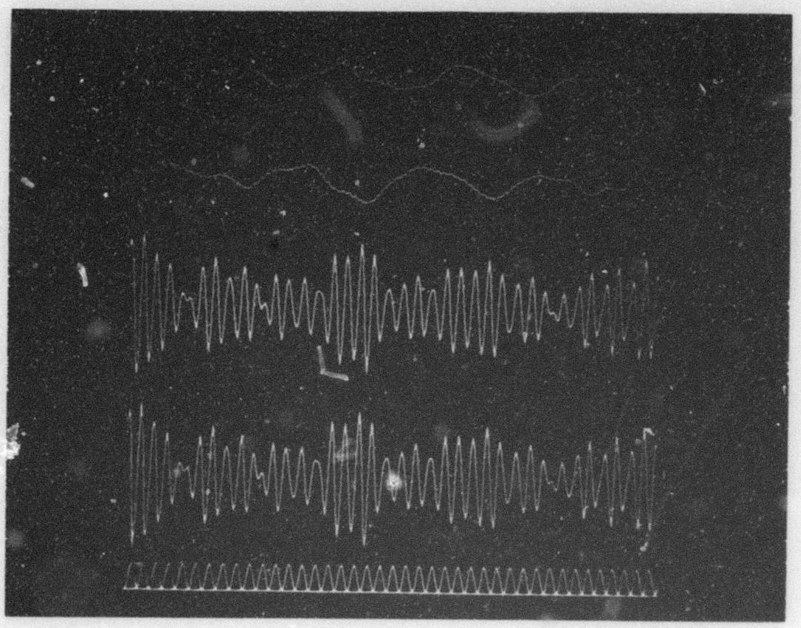


Figure IV.4 Representation of EEG Signal

1. $x_1(N)$, the in-phase component.
2. $x_2(N)$, the quadrature component.
3. The original EEG signal.
4. The reproduced EEG signal synthesized from x_1 and x_2
5. The reference signal.

References

[1] R. V. Floyd and D. C. Lai, "A Model for the Photically Stimulated Electroencephalographic Signals" Tech. Report No. 6740-1, SU-SEL-72-032, Information Systems Laboratory, Stanford University, Stanford, California, August 1972.

[2] R. V. Floyd, D. C. Lai, and J. E. Anliker, "A Model for the Photically Stimulated EEG Signals" Proc. of the 12th Annual San Diego Biomedical Symposium, 1973.

V. EYE-MOVEMENT MEASUREMENT AND TRACKING

1. Measurement of Eye Movement

Setting up an eye-movement measurement and data collection system involves a variety of tasks such as modifying commercial equipment to suit our needs, determining the limitations of this equipment as they affect our experiments, calibration of equipment, and input of data to the PDP-15 computer. At present, our analysis of eye-movement includes computer programs to (1) plot the digitized eye-movement data on either incremental plotter or storage oscilloscope, (2) search through these data and manipulate them for processing in various ways, (3) use pattern recognition techniques to determine "fixations" in the eye-movement data, and (4) label fixations as they are plotted. Other programs provide for more detailed analysis of the eye-movement data to determine the boundary conditions between fixations and saccades with the ultimate goal of predicting in real time the transition from fixation to saccade and vice versa. The techniques for eye-movement measurement and eye-movement data acquisition, and their analysis are described separately as follows.

a. Measurement Techniques and Data Acquisition

i. Measurement Apparatus: The equipment currently used for optical measurements of eye-movements is a commercial unit manufactured by Biometrics, Inc., and dubbed the "Eye-Movement Monitor - Type SG" (EMM). Its principle of operation involves the photoelectric scleral reflection technique (shining infrared light onto

the eye and picking up the reflected light with pairs of photo-electric sensors). Horizontal movements are monitored from the subject's right eye, and vertical movements are monitored from the left eye. This is based on the assumption that both eyes normally move together. To monitor horizontal movements, the sensors detect varying amounts of light reflected from the dark portion (iris) and the white portion (sclera) of the eye as varying areas of these two regions move under each sensor of the pair. The output voltage of the horizontal channel of the EMM is proportional to the voltage difference between the two horizontal sensors.

Vertical movements are measured by monitoring the movement of the upper eyelid, which is proportional to the actual vertical movement of the eye within a degree of accuracy. The vertical sensors are positioned so that vertical movement of the lid results in varying areas of eyelid passing under both sensors. The output voltage of the vertical channel of the EMM is proportional to the sum of the voltages of the two sensors, which is proportional to the area of eyelid reflecting light to the sensors.

The sensors and light sources are mounted on an eyeglass frame worn by the subject. One source and two sensors are mounted on each of two platforms which can be moved independently in three dimensions to position each sensor-source group as desired. A bothersome problem when adjusting sensor-source position in any one dimension is the starting friction which must be overcome because movement is accomplished by sliding a plastic sleeve along

a metal track. If only a small movement is desired, one tends to overshoot the desired position when overcoming the starting friction. Frustration and wasted time rapidly become painful realities. Therefore, a modification has been made to this sensor-source apparatus to facilitate accurate and rapid adjustment. Movement in each direction is now accomplished by turning a small screw (one for each dimension of movement); the mechanism operates similar to a micrometer. This modification has significantly reduced adjustment and calibration problems and has greatly increased efficiency in data collection. A picture of the modified Biometrics glasses is shown in Figure V.1.

The horizontal and vertical output voltages from the EMM are input to separate analog-to-digital conversion channels on the PDP-15, digitized and stored on magnetic tape by a general digitizing program or various special calibration programs. Simultaneously, these voltages may also be recorded on separate channels of analog tape at high speed so the data can later be digitized when played back at a speed slower than the recording speed, effectively resulting in a high sampling rate. The sampling interval for digitizing the EMM data is presently adjustable upwards from a minimum of one millisecond. This is adequate in the sense of the Nyquist criterion since the EMM amplifiers have a maximum bandwidth of 500 Hz. (2 ms.). The basic limitation on sampling rate of our computer system is due to the speed of the magnetic tape storage.

The remainder of the present experimental setup, shown in

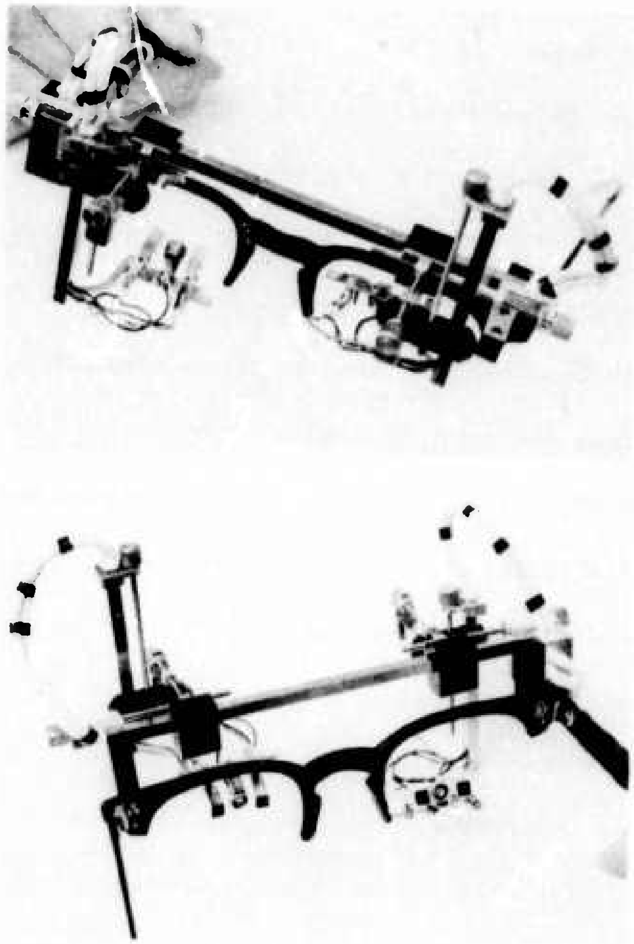


Figure V.1 Modified Biometrics Glasses with Light Sources and Sensors

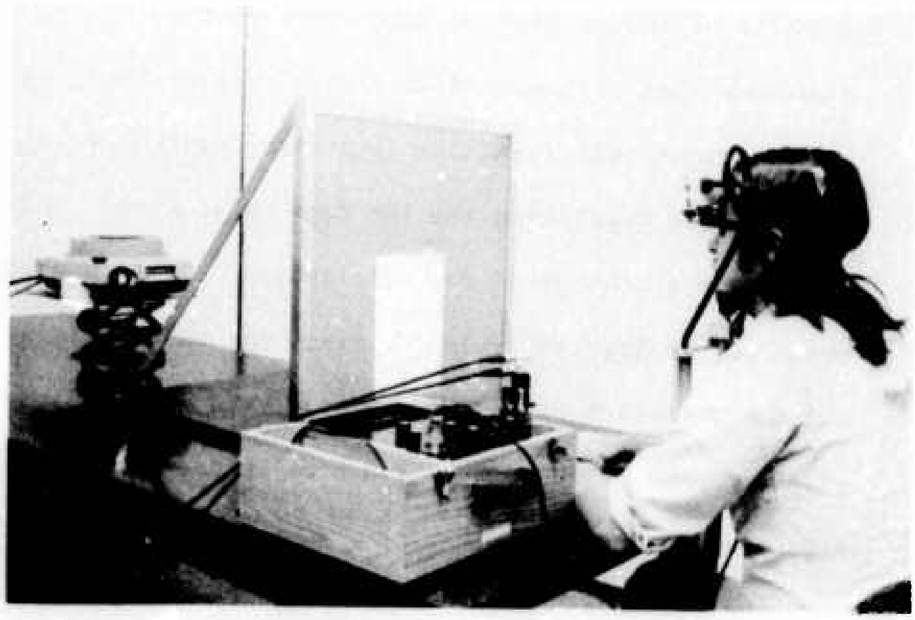


Figure V.2 Experimental Setup

Figure V.2, consists of a headrest, random access slide projector, rear projection screen, and storage oscilloscope. With the EMM glasses in place, the subject's head is secured in a headrest apparatus to minimize head movement, which creates problems that will be discussed later. Visual stimulus patterns are presented to the subject using either (1) a random access projector to project slides onto a rear projection screen in front of the subject, or (2) computer-generated patterns displayed on a storage oscilloscope. Both stimulus systems are controlled by the PDP-15 computer.

A parameter of concern is the resolution of the EMM. Consider the eye fixating on a point of the display. When it fixates on another point at some distance, x, from the present fixation point, it must rotate through an angle equal to the angle at the eye (fovea) subtended by the cord connecting these two points. Resolution is the smallest angle through which the eye moves such that the measurement output voltages of the EMM are detectably different. Alternatively, if the eye-movement data is plotted, the two fixations subtending this angle will be distinct. Resolution was determined by presenting a series of square dot matrices, each succeeding matrix having smaller inter-dot distances, to the subject and directing the subject to fixate these points. When plotted, the distinctiveness of adjacent points can be determined. Smaller and smaller inter-dot distances were presented until the adjacent plotted points could not be distinguished. From the smallest distinctive inter-dot distance and the distance from the

plane of the display to the eye, the angle subtended and thus the corresponding angular movement is determined. This is the resolutuion of the EMM measuring technique and equipment. Horizontal resolution has been determined to be approximately ½ degree and vertical resolution just under 1 degree using the above evaluation technique. These values are close to the manufacturer's specifications and the theoretical limits for this eye-movement measurement technique (scleral reflection).

ii. Calibration: Calibration must be the first step in any experiment in which eye-movement data are collected so that the calibration data can serve as a reference to which the experimental data can be scaled. Our present calibration procedure is as follows:

(1) The subject is directed to fixate on the appropriate points of a square 9-point matrix (3 rows x 3 columns) on the rear projection screen or the storage scope while the experimenter adjusts the sensors so that symmetry of voltage readings (monitored on another oscilloscope) around the center point is obtained for both vertical and horizontal movements. The horizontal and vertical extremes of this matrix subtend an angle of 20 degrees at the subject's eye. A 17-point calibration matrix of two concentric circles subtending angles of 10 and 20 degrees at the eye is also used. The calibration matrices are shown in Figures V.3 and V.5. Eye-movement data obtained by directed fixations on the points

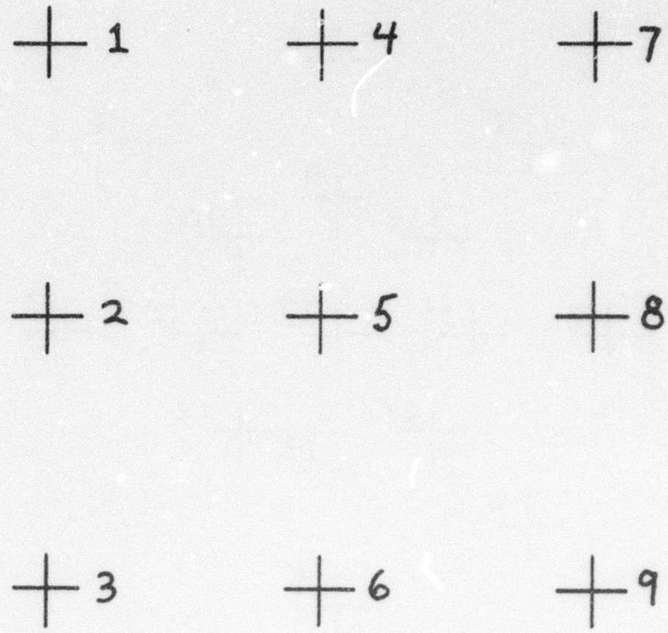


Figure V.3 9-Point Calibration Matrix

Figure V.4 Fixation on Points of 9-Point Matrix

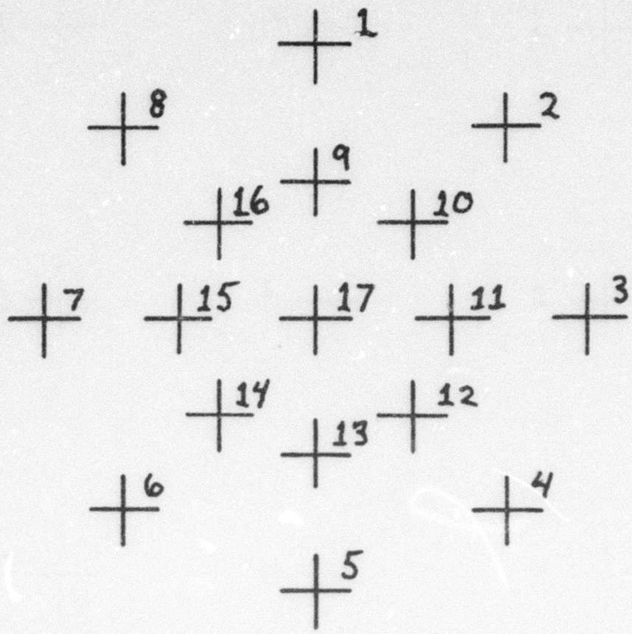


Figure V.5 17-Point Calibration Matrix

Figure V.6 Fixations on Points of 17-Point Matrix

of these matrices are shown in Figures V.4 and V.6, respectively.

(2) When the experimenter obtains what appears to be the best possible adjustment, he checks this by running a calibration program. This calibration program samples the horizontal and vertical EMM voltages as the subject is directed to fixate on the individual points of the calibration matrix. The average voltage for each point is determined and stored.

(3) Various calculations are done on these data and results printed so the experimenter can determine how good the sensor adjustment is. The calculations include horizontal and vertical crosstalk, zero shift, horizontal and vertical full-scale values (indicative of symmetry), and variations of the position voltages about the average values.

(4) If these results are unsatisfactory, the sensors are readjusted appropriately; and the calibration program is rerun. This is repeated until the experimenter is satisfied that he has obtained the best possible calibration.

(5) The desired experiment is performed, with the calibration being checked at various points of the experiment.

Crosstalk and zero shift are the most significant problems encountered in the eye-movement measurements because they result

in inconsistent data and complicate data analysis. Horizontal crosstalk occurs when a non-zero horizontal sensor voltage difference is produced for what is a purely horizontal eye-movement. The effect of this crosstalk is to distort the mapping from the two-dimensional eye-movement space to the two-dimensional voltage space so that this mapping is not a linear one-to-one transformation. Thus, for a bad sensor adjustment, a given voltage reading may represent several eye fixations separated by several degrees of eye-rotation. Clearly, we desire to reduce this distortion as much as possible.

Vertical crosstalk is negligible, but horizontal crosstalk can be severe for different subjects and on different days for the same subject, even with optimal sensor position. This crosstalk is due to the interference of the upper and lower lids with the light received by the horizontal sensors as the eye makes vertical movements. Because of the uneven curvature of the eyelids, interference with each of the two sensors is asymmetrical, resulting in a "noise" voltage difference between the sensor outputs. However, since we know the cause of the distortion, we can adjust the sensors efficiently and quickly so as to minimize the error. To further circumvent this problem, we can use the calibration results to determine a linearization function for the distorted regions and modify the experimental data for this region in accordance with this function. Another approach is to constrain the displays presented to the subject so as to avoid high-distortion areas.

The regions with the greatest amount of distortion include those downward and to the right and upward and to the left in the subject's field of vision, when eye-movements of greater than 5 degrees from the center of the display must be made. Movements within a circle having its center in the center of the subject's field of vision and subtending 10 degrees with the eyes generally yield a very good one-to-one transformation of eye-movement position to sensor voltage. Examples of distortion in the data can be seen in Figures V.4 and V.6.

Zero shift is another problem, since the output voltages of the EMM are nulled (zero volts) while the subject fixates on the center of the display. Movements away from this center point create plus or minus voltages relative to this null. Head movement shifts the voltage output to a non-zero value for a fixation on the center point. Symmetry of left-right, up-down voltages is maintained with respect to the new offset "zero", but we want to reduce this offset drift as much as possible to simplify data analysis. This is done principally by constraining the subject's head in the headrest. We may have to resort to using a bite bar later in some experiments to minimize errors resulting from head movements.

b. Data Analysis

We have developed what is essentially a more automatic version of Noton's [6] scanpath detection procedure. Analysis is performed on data digitized and stored on magnetic tape. A very flexible program

has been written for accessing these data, performing calculations on it, plotting it on the CRT or incremental plotter, etc. This program allows one to access data continuously, stop and restart access at any time (enables one to view plots on the CRT in sections, for example), access a single point, or access any number of sequential points in a record or file.

We currently have different fixation recognition and labeling programs. These were written for use in the experiments described above to facilitate plotting and numbering sequential fixations. The recognition algorithms are based on the simple criterion that some number of sequential sample points be clustered within a certain "area" (X-Y voltage region). The number of points is determined by the time interval between samples, since a fixation is defined for our purposes to be greater than or equal to 200 milliseconds. This value is chosen for two reasons. First, 200 ms is generally acknowledged in the literature to be approximately the minimum duration of fixation (often referred to as the inter-saccadic interval). Secondly, by looking at our own plots of eye-movement data, we can count the number of sample points in clusters which "look like" fixations; from this and the sampling rate, we calculate the minimum fixation duration to be approximately 200 milliseconds.

This is a very rough and somewhat heuristic approach to defining a fixation, or more importantly, the boundary (in time) between a fixation and a saccade. Though adequate for the above experiments, this definition and the algorithms implementing it are not very useful for the types of experiments we hope to be performing in the near future.

Therefore, the present effort involves more accurate determination of the fixation - saccade boundary. This is important because the brain accepts visual data from the retina only during eye fixation, not during saccades. We should like to predict the onset direction and extent of the saccade at the earliest possible moment and to use various parameters of the saccade (such as velocity and acceleration) for predicting the location of the next fixation.

There are indications from previous work by Yarbus [9] that certain saccade and fixation parameters vary characteristically during the time course of a saccade or fixation. Specifically, Yarbus noted the magnitude of the velocity and acceleration of saccades very nearly approximated a sinusoidal function with respect to time. While these results may be partially attributable to artifacts in Yarbus' measurement techniques, the suggestion that velocity and acceleration follow a time course having a maximum value with perhaps some symmetry about this maximum is reasonable from a physiological point of view of the saccadic meachanism. If we can find a similar characteristic pattern of velocity and acceleration, we can use these characteristics to predict not only the onset of a saccade, but also its direction and terminating fixation position. We are looking for a consistent pattern of parameters.

Initial work on this problem is being done off-line to determine which characteristic patterns of what parameters exist. The next step will be to extract the parameters providing the most important information and implement their analysis in a real-time environment.

In summary, we have been developing techniques for collecting and

analyzing eye-movement data for the purpose of discovering spatial cues that can be used as keys for memory retrieval. Having started with off-line analytical procedures, we are now designing algorithms for real-time techniques for the extraction of the critical eye-movement parameters and using this information in feedback control schemes to influence impression and recall of visual images.

2. A Model for Eye-Movement Tracking

In order to feedback the spatial cues, it is necessary to have an estimator and predictor filter for the eye-movement process. Such a filter is used in tracking the eye movement and eye fixation. This information is used to generate visual stimuli at appropriate spots in the visual field.

For estimation and prediction, it is useful to have a model of the eye-movement process. At present, some models have appeared in the literature, but most of these are concerned with eye-tracking movements [1, 2]; i.e., movements of the eye when following a moving target, and eye-movement control [3, 4]. However, for the study of the interaction between the memory and eye movements, we need a model for viewing stationary patterns. Recently, a few such models have been proposed [5, 6, 7, 8]. These have not, however, been used for the purpose of prediction. We shall develop a model which can be used for this purpose. We shall first give a brief account of the existing theories.

A theory of pattern perception has been propounded by Noton and Stark [5, 6]. This theory is concerned with explaining eye movements

during learning, subsequent recognition, and recognition under unfavorable real world conditions (e.g., distortion, noise, etc.) of visual patterns. The assumption is made that the learning of a pattern involves constructing an internal representation of the pattern in memory while recognition involves a search for an internal representation which matches the given pattern. Noton and Stark proposed a particular form of the internal representation and of the process of finding and matching. They observed that for a particular subject viewing a particular pattern, there exists a fixed sequence of fixation points which occurs for a large fraction of the inital viewing time. They call this sequence a scanpath. The scanpath is different for different subjects and for different patterns (although for a particular pattern, the "features" are the same for most subjects). At the time of the initial viewing, about 35% of the time was spent in repeating the scanpath with irregular fixations during the rest of the time. This scanpath was found to be repeated in 65% of later viewings of the same pattern.

Based on these observations, Noton and Stark proposed that the internal representation of an object in memory is primarily serial in nature. It consists of a network of memory traces, termed a "feature network", which records the features of the pattern and the movements required to go from feature to feature in the pattern. During learning, this representation is formed. During recognition, the features of the given pattern are matched sequentially, with the internal representation directing attention to the successive features of the patterns and thus causing the eye-movements. Such a model would explain the

occurrence of scanpaths, which are supposedly due to the force of habit in perception so that each person develops his own characteristic path for a pattern.

This is a very simple model which is not very useful for prediction. It can be extended to a statistical model, where the scanpath is the most probable direction of eye movement, but there is a probability assignment for the other parts of the feature network also. This can be used for prediction using standard statistical methods. It is an empirical model, for which the scanpath has to be determined first.

The above model does not give any quantitative way of determining the scanpath in advance, but this difficulty can be remedied by the model proposed by F. J. Prokoski [7]. This model is a non-linear spatio-temporal control system with the pattern as the input and a series of vectors representing the eye movements as the output. Temporal and spatial sampling is assumed so that the system is a discrete time control system. The image presented to the eyes is first mapped into the receptor response. The mapping function is the relative sensitivity of the receptors as a function of the gaze angle. Now a mapping is made to a short-term memory. It is assumed that the short-term memory decays exponentially with time and the receptor response is added on directly to the contents of this memory. The rate of decay is one of the parameters of the model and depends on the subject.

It is assumed that the visual cortex performs an absolute value difference cross-correlation between the contents of the short term memory and the receptors response at any given time. This cross-correlation is restricted to the foveal portion of the receptor array.

The resulting function determines where the eyes will be moved at the next sampling instant. The eyes are moved if the current intensity sum over the foveal area falls below a certain threshold, determined by the short-term memory and a parameter representing the psychological "set" of the subject. In that case, the eyes are moved by the control vector determined by the minimum and the origin of the output of the cross-correlation.

The assumptions made in this model are to some extent supported by physiological and psychological data. The model has been used for prediction of teaching movements. With a few extensions and modifications, it can be used for our purpose. For the case of recognition, long term memory has to be incorporated in the model. Also, the model at present is deterministic and produces a specific output vector. We can change it to a stochastic control system where the output vector is a random function of the cross-correlation array. This will be helpful in determining the accuracy of the predictive filter in a quantitative sense. The model can be useful at least for the initial production of the scanpath.

Another model to be considered is a stochastic model [8] which, as given, applies only to eye movements during fixation and not for saccadic movements between fixations. Within a fixation, these are "local areas" of fixation and transitions from one local area to another. Theoretical transforms of the probability density function for local area motion and the moments of the number of transitions in a given time T are obtained. This model can probably be extended to cover transition from fixation to fixation. By looking at the fixation

data, we should be able to predict when the fixation ends.

The above summarizes the various types of models we will be considering for our use. Extensions and modifications to any one or more of these can make them suitable for our purpose of estimation and prediction.

Now we will briefly describe the work on eye movement data collection and analysis which has been accomplished. The instrument used for measuring eye movements uses the scleral reflection technique modified to measure vertical and horizontal movements. The operation of the instrument, the data gathering programs and the calibration methods have been describe elsewhere in this report. Our initial efforts have been directed toward verifying some of Noton and Stark's experiments on scanpaths. Noton and Stark slowed down the eye-movement data, plotted it and marked the fixation points by hand. An algorithm for determining the fixation points is implemented on the PDP-15. A brief description follows.

Raw data are read off the magnetic tape. Calibration data, obtained in the same experiment, are used in processing these raw data. The calibration assumes linearity although different reference values are used for each of the four quadrants. This is done to take care of the crosstalk and the asymmetry in the calibration data.

After calibration, the fixation program has essentially two parts: a) determining when the fixation mode has been entered, and b) once a fixation is decided upon, determining when it ends. Both use certain parameters which are largely empirical in nature. For determining a fixation, the program checks out whether, of N points, M lie within a

certain distance D of the mean value. The values of N, M, and D have been set from reported studies [9] on duration and spread of fixations and from our own observations. To determine the end of the fixation, the program checks if a certain number of consecutive points fall outside of a prescribed mean. Again, the parameters have been set as above.

The fixation points are plotted and numbered and their locations printed. Thus we can automatically determine the sequence of fixations and observe the scanpath.

The performance of this program has been tested by comparing it with the determination of fixations by looking at the plotted data and marking the apparent fixation points. It has been found to be very satisfactory. Some examples are shown in Figures V.7, V.8, V.9, and V.10.

In addition to this, we have programs for plotting data, both on the CRT and the CalComp plotter, for analysis of fixation points (standard deviation, frequency histograms of length of movements, distribution of angle of movements, etc.), for separate plotting of the horizontal and vertical movements and their analysis, etc. Other programs are being written to facilitate the task of data analysis and model building.

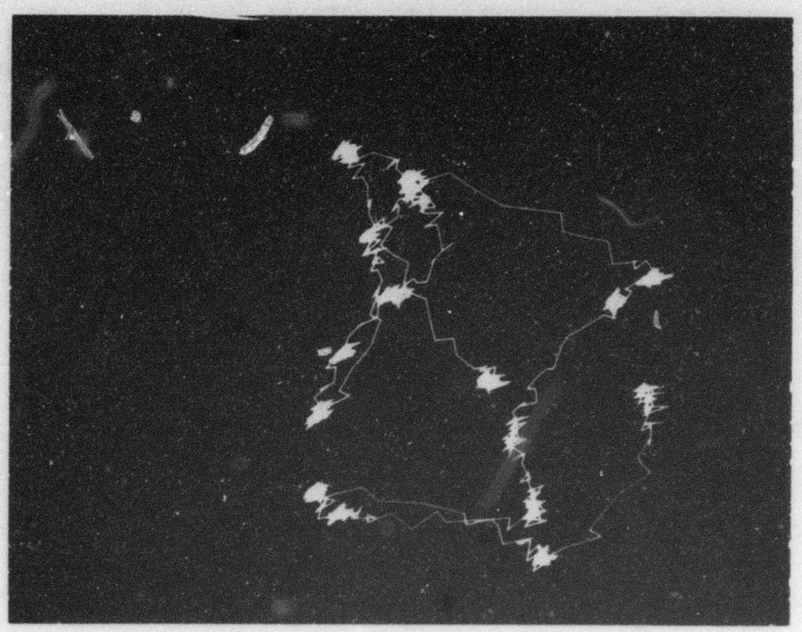


Figure V.7 A plot of eye movement during examination of a simple geometric pattern (10 sec.)

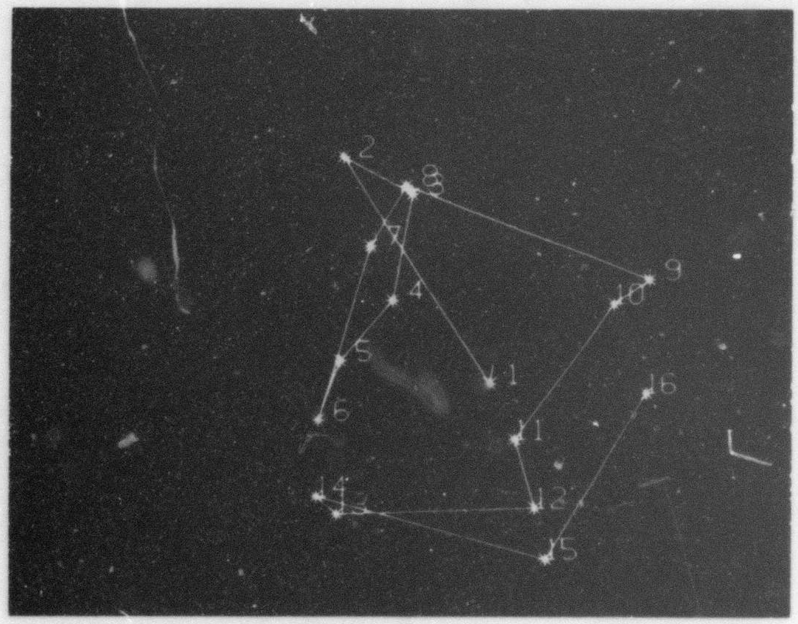


Figure V.8 The sequence of fixations in the above as determined by the computer program

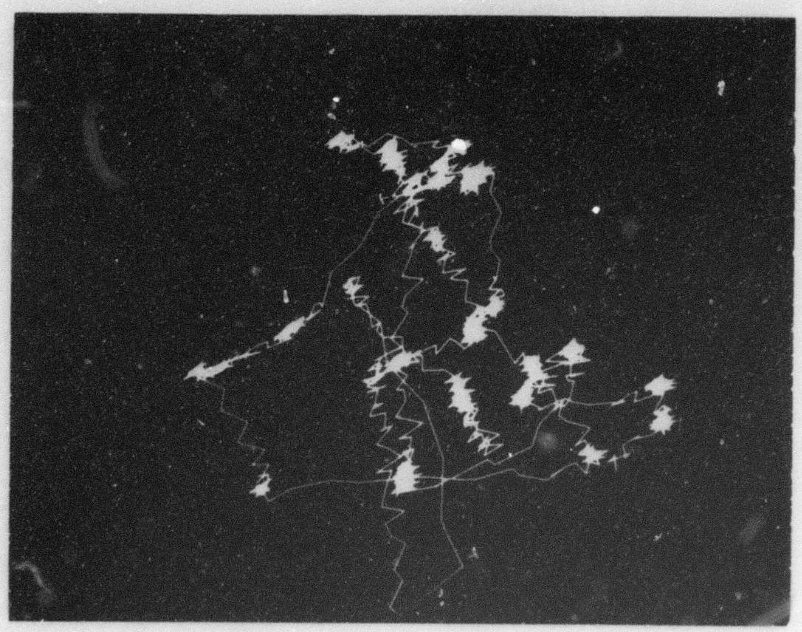


Figure V.9 Plot of eye movement
for another pattern (20 secs.)

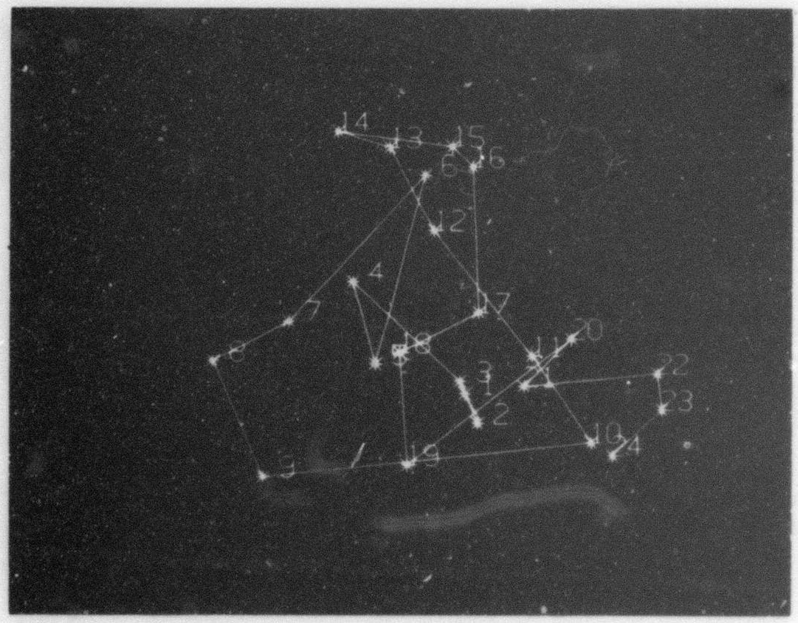


Figure V.10 The sequence of fixation in the above

REFEFENCES

[1] L. Stark, G. Vossius and L. R. Young, "Predictive Control of Eye Tracking Movements", IRE Trans. on Human Factors in Electronics, HFE-3, 52-57, September 1962.

[2] L. R. Young and L. Stark, "Variable Feedback Experiments Testing a Sampled Data Model for Eye Tracking Movements", IEEE Trans. of the Professional Tech. Group on Human Factors in Electronics, HFE-4, 38-51, 1963.

[3] G. Vossius, "The Control of Eye-Movement", Proc. of the 1972 Int'l Conference on Cybernetics and Society, 27-31.

[4] D. A. Robinson, "Progress in Models of Eye Movement Control", Proc. of the 1972 Int'l Conference on Cybernetics and Society, 14-24.

[5] D. Noton and L. Stark, "Scanpaths in Saccadic Eye Movements While Viewing and Recognizing Patterns", Vision Research, Vol. II, 929-942, September 1971.

[6] D. Noton, "Two Models of the Brain with Applications to Artificial Intelligence", Doctoral dissertation, University of California, Berkeley, California, 1970.

[7] F. J. Prokoski, "Modeling the Human in Visual Tasks", Proc. of the 1972 Int'l Conference on Cybernetics and Society, 44-54.

[8] R. Vasudevan, A. V. Phatak and J. D. Smith, "A Stochastic Model for Eye Movement During Fixation on a Stationary Target", Technical Report No. 71-23, Dept. of Electrical Engineering, University of Southern California, Los Angeles, California, May 1971.

[9] A. L. Yarbus, Eye Movements and Vision, New York, Plenum Press, 1967.

VI. CONCLUSION

The work has been proceeding along the lines that we proposed. We have brought up the new PDP-15 computer system to full operational status. We have investigated and implemented some real-time monitoring and prediction schemes and some modeling techniques for EEG and eye-movement variables. For instance, we have implemented a system using a discrete phase-locked-loop that makes possible real-time tracking of amplitude, frequency, and phase variables in the EEG signals and we have implemented an algorithm for tracking the visual scanpath and automatic marking of the fixation points in the scanpaths. Since the inception of this project, we have had several papers published and some are scheduled to be presented/published. During the next reporting period we shall concentrate on the acquisition of experimental data.

By using the continuously up-dated information concerning the eye position and brain state for adjusting the stimulus parameters, we will attempt to enhance the vividness and persistence of the desired after-images. To be more specific, we plan to accomplish the following tasks in the coming year:

1) Develop real-time methods for relating eye-movements to visual targets; further, record the scanpaths for subsequent computer-aided image-enhancement.

2) Analyze the perceptual consequences of visual target presentation that is contingent upon α-phase; analyze the effects upon imagery of phase-contingent blank-field probing.

3) Develop methods for the estimation of eye-pointing based on the combination of optical eye-tracking measurements and electrooculographic measurements so as to achieve a best single estimator for both eye-open and eye-closed conditions.

4) Develop real-time strategies for the control of visual target impression and visual image (memory) persistence and/or recall ability.

APPENDIX A: GS-15--A GRAPHICS SOFTWARE SYSTEM

1. Introduction

GS-15 is a generalized graphics software system for the PDP-15 with RSX-PLUS. The system provides general symbolic and numeric capabilities that can be combined to perform any practical graphics function.

a. System Configuration

In order to use GS-15, a PDP-15 with RSX-PLUS is required. However, since a major portion of the software is coded in FORTRAN, the system could be transferred (albeit with effort) to another machine. GS-15 is currently using the FP15 floating point hardware option, but could more than survive in a non-floating point environment. In addition, our real-time laboratory also includes a VP15A storage scope and a CalComp incremental plotter.

b. Design Objectives

i. LUN Orientation: A primary design consideration was to create a system that was a logical unit (LUN) rather than device oriented. Past experience demonstrated the disadvantages of device-specific graphics software. Furthermore, we wanted a system that would blend unnoticed into RSX.

ii. Extensibility: The other main consideration was extensibility. Since our project's goal is enhancement of human memory and not systems software development, we knew we would not have time to

create a complete graphics system. Instead, the decision was made to create the nucleus of such a system.

For this reason, GS-15 does not include large conglomerate user subroutines (e.g., a single subroutine to draw a complete plot including labeled axes, tick marks, etc.), but instead contains the essential, elementary capabilities found in the intersection of currently available graphics systems. Also included is the capability to easily add additional graphics functions (FORTRAN subroutines) as needed by the user. Furthermore, as a consequence of LUN oriented I/O, all that is required for GS-15 to utilize newly acquired graphics hardware is the writing of another device handler. Once the handler is installed, previously written graphics tasks may immediately use the new device.

2. Overview

As shown in Figure A.1, GS-15 consists of three major system components: the graphics pool, subroutines, and device handlers.

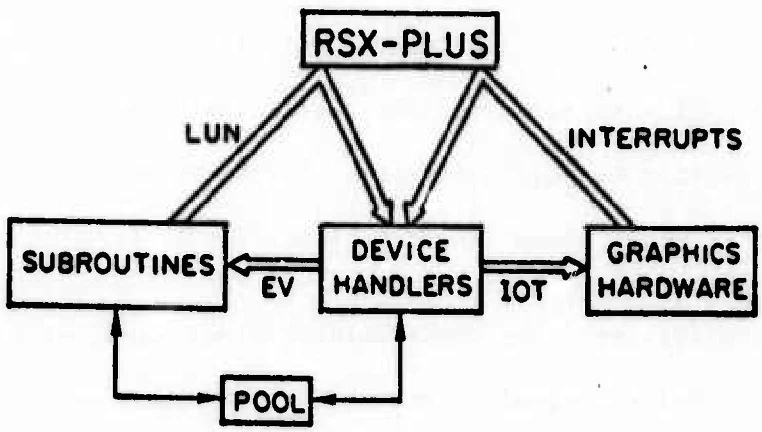


Figure A.1 GS-15 Data Flow

In order to produce a plot, the programmer must first create a task (program) containing calls to one or more GS-15 subroutines. Next, the appropriate LUN should be assigned to a graphics device and the task requested.

When the task CALLS a graphics subroutine, an I/O request node is queued to the appropriate logical unit. All necessary IOTs are executed as soon as the device handler obtains control. When the device handler receives the last interrupt signifying the completion of the particular graphics function requested, the caller's event variable is set to plus one (assuming no errors were detected). All during this process, the pool is referenced and updated--always reflecting the current state of the graphics job.

a. POOL

The graphics POOL (a FORTRAN array) is an integral part of GS-15. Parameters that are common to many subroutines are collected together in the POOL (see Table A.1) rather than being specified repetitively in each subroutine call. Typical elements of the POOL include the graphics LUN (needed by all subroutines), character height and width (needed by all symbolic subroutines), and scaling factors. In addition, certain POOL elements contain information on the internal state of GS-15 (e.g., current location of the plotter pen).

Another advantage of the POOL is maintaining upward compatibility of the GS-15 software. For example, a user might want a subroutine extended slightly (e.g., allowing multiple character fonts) necessitating the addition of another parameter. With GS-15, the new feature

could be implemented by creating a new POOL element (font type) and changing all symbolic routines to recognize this new element. Since no calling sequences would be changed, the modification would be upward compatible.

#	NAME	DESCRIPTION
1	GLUN	Graphics LUN
2	GFNAME	Graphics file name
4	DNSITY	Line density
5	SPEED	Speed factor
8	XMIN	Subject space
10	YMIN	
12	XMAX	
14	YMAX	
16	PXMIN	Object space
18	PYMIN	
20	PXMAX	
22	PYMAX	
24	XFACTR	Internal constants
26	YFACTR	
28	PUPD	Pen up/down flag
29	CPX	Current pen position
30	CPY	
33	STHETA	Symbolic orientation
35	CTHETA	
37	HEIGHT	Sumbolic height
38	WIDTH	Symbolic width

Table A.1 Graphics POOL Contents

b. Subroutines

The second major component in the GS-15 system is a set of (FORTRAN/MACRO callable) graphics subroutines which provide the user interface to GS-15. Their purpose is to make available a set of elementary graphics functions that the user can combine into larger, special purpose graphics routines which might, for example, draw poly-

gons or three-dimensional figures. These elementary functions are realized by queuing one or more I/O request nodes to the graphics LUN specified in the first POOL element. When all such I/O requests have been completed, the event variable specified in the subroutine CALL is set to indicate that the desired function has been performed.

The last formal parameter in most GS-15 subroutines is an event variable. If an event variable is not specified in a graphics subroutine CALL, the subroutine will RETURN to the caller only after the requested function has been completed. On the other hand, control is returned immediately if an event variable is present in the subroutine CALL. By later checking the status of the event variable with the WAITFR system directive, asynchronous operation can be achieved.

Table A.2 lists the subroutines available in the first version of GS-15. It should be kept in mind that the action performed by a subroutine depends on the current assignment of the graphics LUN. For

NAME	FUNCTION
GBEGIN	Initialization
GEND	Termination
GLINES	Connect "N" points
GMOVE	Pen movement
GNEXT	Erase scope/advance paper
GNUMBR	Draw formatted numbers
IGPX, IGPY	Translate subject to object space
GSCALE	Define subject and object spaces
GSET	Modify POOL elements
GSYMBL	Draw single character
GTEXT	Draw groups of characters

Table A.2 GS-15 Subroutines

example, GNEXT erases the scope if GLUN is assigned to the storage CRT, or advances the plotter paper beyond the furthest point drawn in the positive "X" direction if assigned to the plotter.

c. Device Handlers

Graphics device handlers enable GS-15 to control the graphics hardware. They are the recipients of the I/O request nodes queued by the graphics subroutines. IOTs are issued by the handlers based on graphics commands contained within the request nodes. In order to be effective in the RSX environment, the handlers recognize most system I/O function codes (ABORT, ATTACH, etc.) in addition to the non-standard codes needed by the GS-15 system.

Currently, two graphics device handlers exist: TV, which controls a VP15A storage scope, and PL, which controls a CalComp incremental plotter. Both were written in MACRO-15 and occupy less than 2000_8 words of memory. Since TV and PL are very similar, only one source file (TVPL) is maintained with conditional assemblies producing the appropriate binary files.

A new handler (GF) will be added shortly. Assigning GLUN to GF will cause a graphics file (whose name is in the second POOL element) to be created on disk or dectape depending on the current assignment of LUN 17 (normally assigned to disk). Once the file has been created, it may be processed by a low priority background task and/or saved for future use. This feature will provide the fastest mode of operation for GS-15. With it, "snapshots" of real-time processes may be stored on the disk and later selectively viewed.

3. Usage

a. Task Structure

As mentioned previously, a graphics task contains calls to various GS-15 subroutines. The general structure of a graphics task is shown in Figure A.2

FORTRAN declarations

CALL GBEGIN

FORTRAN statements and CALLS to GS-15

CALL GEND
STOP
END

Figure A.2 Typical Graphics Task

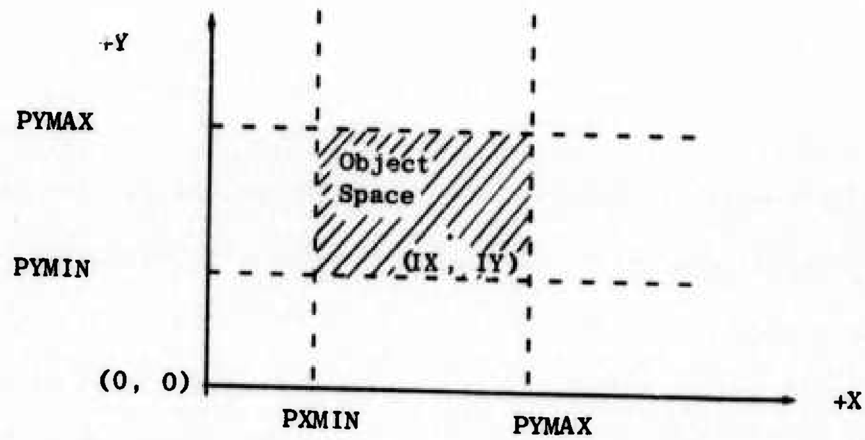
GBEGIN must be the first GS-15 subroutine called, and may be called as many times as desired. Its purpose is to initialize both the POOL and the currently assigned graphics device handler. Although not a requirement of GS-15, GEND should be called before the task terminates in order to allow GS-15 to perform certain "cleanup" procedures.

b. Subject/Object Spaces

A common problem in computer installations with graphics facilities is adapting "real data" to the addressing schemes used by graphics hardware devices. For example, a scientist might want to draw the graph of an electrical waveform on a CRT. The voltages corresponding

to the actual waveform would take on real values between, say, -10 volts and +10 volts (e.g., 1.625, -8.301, 0.108, etc.). The CRT, however, requires a pair of integers to address a particular point on its display surface (e.g., the Tektronix 611 storage scope contains a 1024 x 1024 grid of such points). The problem is to create a simple mechanism in the graphics system to convert real valued data (which, in absolute value, could approach 10^{75}) into integer data (in the range 0 to 1023). GS-15 solved this problem by utilizing subject and object spaces.

i. Object Space: As shown in Figure A.3, the object space is a parameterized subset of the first quadrant of a discrete euclidean two-space. Note that a general point (IX, IY), in the



$$\text{PXMIN} \le \text{IX} \le \text{PXMAX} \qquad \text{IX, IY} \in \{0, 1, 2, \ldots\}$$
$$\text{PYMIN} \le \text{IY} \le \text{PYMAX}$$

Figure A.3 Object Space

object space, is defined by an ordered pair of non-negative integers. Furthermore, the object space may be redefined by

changing (see GSCALE below) the values of PXMIN, PYMIN, PXMAX, and PYMAX.

The purpose of the object space is to define where on the graphics device the user wants plotting to be done. For example, to utilize the full face of a Tektronix 611 CRT, PXMIN and PYMIN should be set equal to 0 (zero) and PXMAX and PYMAX should be set equal to 1023. Other uses of variable object spaces are described below (see next section).

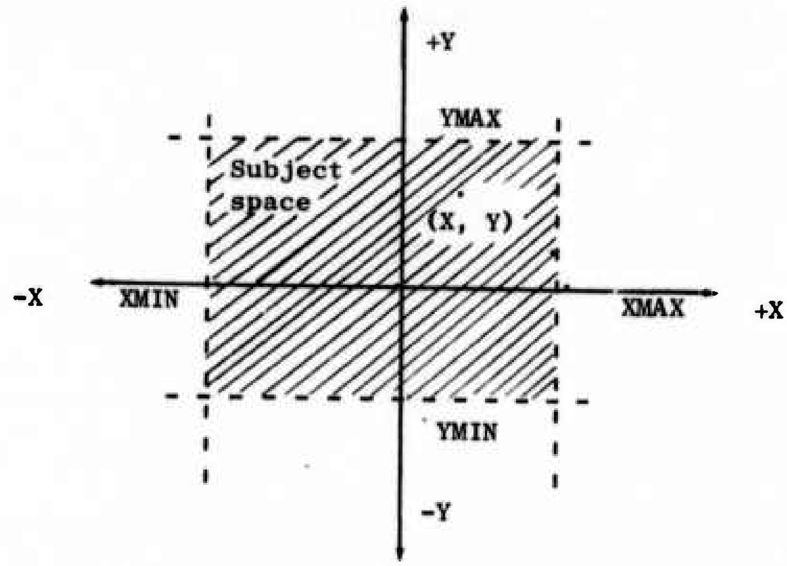
The discrete two-space of which the object space is a subset is identical to the address space present in all incremental (i.e., digital) graphics devices. In other words, most devices allow the user to either connect (plotter) or illuminate (CRT) any of a number of points in an imaginary grid on the device. That portion of this "grid" that is of interest to the programmer is precisely the object space. GS-15 assumes that the object space is wholely contained within the usable address space of the graphics device.

ii. Subject Space: As mentioned earlier, actual data does not lend itself to direct expression in the object space. For this reason, the subject space was created. The subject space is a subset of the normal (non-discrete) euclidean plane (see Figure A.4). Unlike object space coordinates, points in the subject space are specified by pairs of <u>real</u> numbers. This subject space may also be defined by the user through use of the GSCALE subroutine.

Normally, the subject space is defined such that it encloses all the data points that are to be displayed. For example,

assume a plot of SIN(θ) is desired for $-\frac{\pi}{2} \leq \theta \leq \pi$. In this case, the subject space could be defined as

$$XMIN = -\frac{\pi}{2},\ XMAX = \pi,\ YMIN = -1, \text{ and } YMAX = 1.$$



$$-10^{75} \leq XMIN \leq X \leq XMAX \leq 10^{75}$$
$$-10^{75} \leq YMIN \leq Y \leq YMAX \leq 10^{75}$$
$$x, y \in \mathbb{R}$$

Figure A.4 Subject Space

iii. Subject/Object Mapping: If by chance the data to be plotted is integer-valued and within the object space (e.g., a graph of time vs. digitized analog signals), the plot may be produced without the need of the subject space. On the other hand, suppose that the data points to be plotted are not contained in the object space (a more likely situation).

GS-15 contains two integer-valued FORTRAN functions (see IGPX and IGPY below) that define a mapping or projection of of the subject space onto the object space (i.e., onto a portion

of the graphics device). This mapping performs the necessary scaling and translation from real-valued data to the integer-valued points in the object space (see Figure A.5).

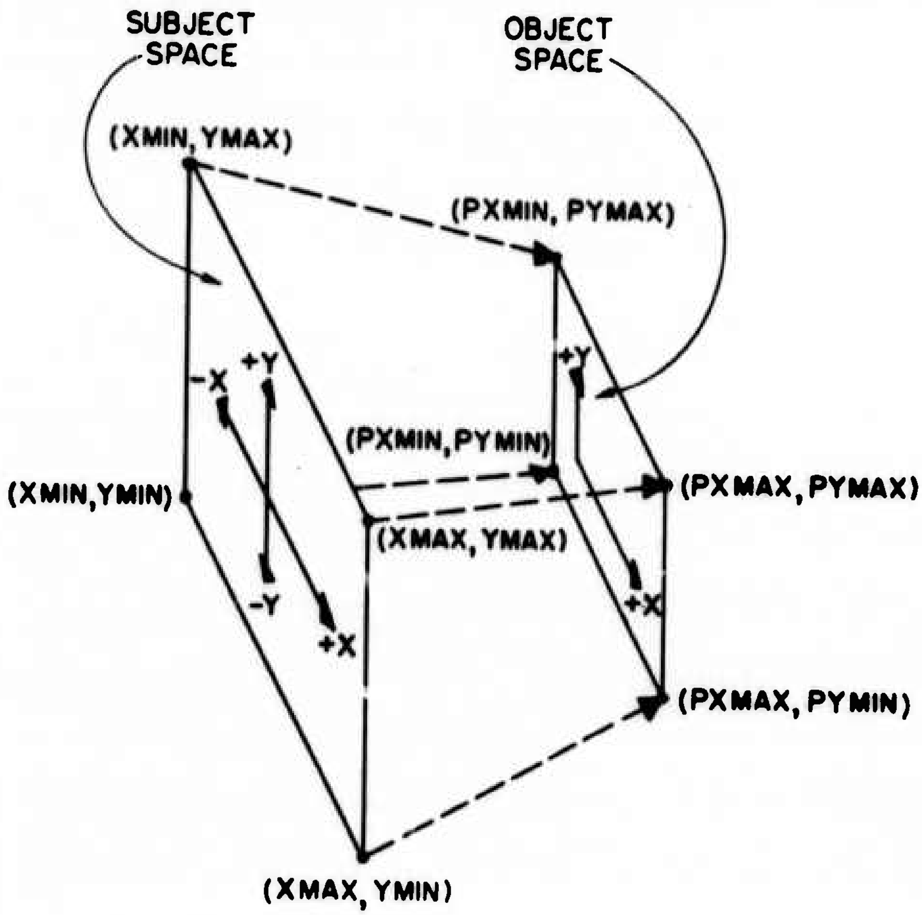


Figure A.5 Subject/Object Space Projection

Currently, most GS-15 subroutines are oriented towards the object space. One exception is GLINES which will connect together "N" points in the subject space. In the future, other subject space oriented routines will be written. It should be pointed out that subroutines that expect object space coordinates <u>can</u>

work with subject space coordinates by using IGPX and IGPY.

The use of the subject/object space mapping has advantages that may not be apparent. One very useful device is that of linearly expanding or contracting a picture by changing the subject space definition. In addition to linear translations, rotations about one or both axes can be performed. For example, subject space data may be rotated about the "X" axis by interchanging the values of YMIN and YMAX.

c. Symbolic and Numeric Modes

GS-15, like most graphic systems, has two modes of operation. The first and most frequently used mode allows the user to draw lines, curves, etc. This mode (numeric or line drawing) is used to plot data and draw figures (one user has drawn a helicopter in flight). In fact, numeric mode subroutines (GMOVE, GLINES) allow the programmer to have complete control of the hardware. For example, GMOVE (when GLUN is assigned to PL) is able to raise/lower the plotter pen and move the pen to any position that the hardware can reach. Subroutines that expect subject space parameters usually operate in numeric mode.

In symbolic mode strings of characters may be drawn whose height, width and orientation depend on values in the POOL. Because textual annotations are usually placed at specific locations on the graphic device (e.g., 3" from bottom of CRT and centered), symbolic subroutines (GSYMBL, GTEXT) are oriented towards the object space.

d. POOL

As mentioned earlier, the POOL is a vector of parameters common

to many subroutines. A bri f explanation of the POOL elements follows.

i. GLUN: This element contains the logical unit number (LUN) on which GS-15 I/O request nodes are queued (see GQUEUE subroutine). The default value of GLUN, 23, corresponds to the system graphics LUN.

ii. GFNAME: When the graphics file device handler (GF) is operational, I/O request modes may be written into a file whose name is in GFNAME by assigning GLUN to GF. This file will be written on the device corresponding to LUN 17 (either DECtape or disk).

iii. DNSITY: DNSITY defines the number of points per inch in lines drawn by GS-15. The CRT, for example, need not illuminate every point in order to draw well-defined lines. Decreasing the DNSITY can, therefore, speed up execution without creating a noticeable loss in quality.

iv. SPEED: This element allows the user to conveniently cause an additional delay between the connecting/illuminating of points. SPEED = 0 will cause no dealy. SPEED ≠ 0 will cause a delay of "SPEED" number of clock ticks (on our machine 1 tick = 1 msec) between points.

This option has two uses. First, for many applications, the CRT will draw lines too quickly. Setting SPEED to non-zero allows the displaying to be done as slow as the user wishes. A second advantage appears when RSX is being used in a multi-programming

mode (i.e., more than one task executing simultaneously). Since the delay is implemented using the MARK and WAITFR system directives, the processor may be used by other tasks during the delay period. The processor may not be used by another task when running at default SPEED (0) since the CRT graphics functions are not interrupt driven.

v. XMIN-PYMAX: These eight values define the current subject and object spaces. They may be changed through use of the GSCALE subroutine.

vi. XFACTR/YFACTR: In order to speed up the calculation of the mapping function (see IGPX and IGPY subroutines) between the subject and object spaces, these internal constants are computed and kept in the POOL. They are changed whenever GSCALE is called.

vii. PUPD: PUPD keeps track of whether the pen is currently up (=1) or down (=0).

viii. CPX/CPY: These elements contain the current location of the pen within the object space.

ix. STHETA/CTHETA: Theta is defined as the symbolic orientation angle (in degrees). STHETA and CTHETA contain respectively the SIN and COS of theta. Theta is considered to be element 33 in the POOL and may be set by using GSET (see below).

x. HEIGHT/WIDTH: These elements define (in inches) the size of characters drawn by GSYMBL and GTEXT.

e. Subroutines

Following is a description of all user subroutines available in GS-15. Other subroutines exist (see section on implementation) that are used internally by GS-15.

GBEGIN

Parameters: None

Function: Initialize the POOL to default values (as shown in Table A.3). Initialize the device handler currently assigned to GLUN. Must be first GS-15 subroutine called. May be called as many times as necessary without intervening GEND.

POOL Element	Default Value
GLUN	23
GFNAME	"GFILE"
DNSITY	100
SPEED	0
XMIN	-100.0
YMIN	-100.0
XMAX	100.0
YMAX	100.0
PXMIN	0.0
PYMIN	0.0
PXMAX	1023.0
PYMAX	1023.0
STHETA	0.0
CTHETA	1.0
HEIGTH	.233"
WIDTH	.166"

Table A.3 POOL Default Values

GEND

Parameters: None

Function: Perform GS-15 termination procedures. No further graphics operations are allowed until GBEGIN is re-entered.

GLINES (I, N, X, Y [, IEV])

Parameters:

Parameters:	I	Denotes initial point in X and Y
	N	Number of points to connect
	X	Real vector of abscissa values in subject space
	Y	Real vector of ordinate values in subject space

Function: Starting at (X(I), Y(I)), connect "N" points in the subject space to produce "N-1" line segments.

GMOVE (IX, IY, JNJ[, IEV])

Parameters:	IX, IY	Object space location
	JNJ	Join/No join flag.

Function: Move the pen from the current location to (IX, IY). If JNJ = "JOIN", draw a line between the two positions. If JNJ = "NOJO", move pen to (IX, IY) without drawing a line. Both IX and IY must be ≥ 0.

GNEXT

Parameters: None

Function: If GLUN is assigned to TV, erase the CRT. If assigned to

PL, move plotter paper beyond furthest point drawn in "+X" direction.

GNUMBR (IX, IY, FORMAT, VALUE [, IEV])

Parameters: IX, IY — Object space location

FORMAT — Format specification

VALUE — Value to be displayed

Function: GNUMBR plots a formatted number on the current graphics device. As in GTEXT, IX and IY are the object space coordinates of the leftmost character in the numeric field. FORMAT is a <string> of from 1 to 5 characters that defines a FORTRAN format specification (e.g., 'I7', 'F10.6', 'E8.6', 'D16.9', 'Ø6'). Note that (IX, IY) specifies the position of the leftmost character in the <u>field</u> which may or may not be the leftmost non-blank character.

The number or expression to be displayed is the fourth parameter. It may be INTEGER, REAL, DOUBLE PRECISION, or LOGICAL. It may <u>not</u> be DOUBLE INTEGER. As in a WRITE statement, the type of VALUE should match the type implied by the format specification.

IGPX(X)

Parameters: X — Abscissa value in subject space

Function: IGPX is a function that returns an integer value equal to

the object space equivalent of X. The mapping function is defined by

$$IGPX(X) = \frac{PXMAX(XMIN-X) + PMIN(X-XMAX)}{XMIN - XMAX}$$

IGPY(Y)

Parameters: Y Ordinate value in subject space.

Function: Like IGPX, IGPY returns the object space equivalent of Y. The mapping function is defined by

$$IGPY(Y) = \frac{PYMIN(YMAX-Y) + PYMAX(Y-YMIN)}{YMAX - YMIN}$$

GSCALE (XMIN, YMIN, XMAX, YMAX, PXMIN, PYMIN, PXMAX, PYMAX [, IEV])

Parameters:

$$\left.\begin{array}{l}\text{XMIN}\\\text{YMIN}\\\text{XMAX}\\\text{YMAX}\end{array}\right\}\ \text{New subject space}$$

$$\left.\begin{array}{l}\text{PXMIN}\\\text{PYMIN}\\\text{PXMAX}\\\text{PYMAX}\end{array}\right\}\ \text{New object space}$$

Function: Redefine subject and/or object spaces. XMIN, ..., PYMAX must be floating point quantities.

GSET (I, VALUE [, IEV])

Parameters: I Pool index (see Table A.1)

VALUE New POOL element value

Function: Set POOL element "I" equal to VALUE. VALUE must be a

floating point quantity.

GSYMBL (IX, IY, ICODE [, IEV])

Parameters: IX, IY — Object space location

ICODE — Symbolic code (see Table A.4).

Function: Draw the symbol corresponding to ICODE at (IX, IY). The size and orientation of the symbol will be determined by the POOL (see Table A.1). Symbols with ICODE $< 40_8$ will be drawn centered about (IX, IY).

GTEXT (IX, IY, N, TEXT [, IEV])

Parameters: IX, IY — Object space location

N — Number of characters

TEXT — Characters to be drawn

Function: Draw the first "N" characters of TEXT at (IX, IY). TEXT may be of three forms: a FORTRAN ⟨string constant⟩ , ⟨string variable⟩, or an ⟨array element⟩. Note that TEXT <u>may not</u> be an ⟨array name⟩. Following are examples of the three forms:

```
Call GTEXT (0, 100, 4, 'TEST', IEV)
Call GTEXT (512, 512, 2, CHARS, IEV)
Call GTEXT (0, 0, 17, BUFFER(3), IEV)
```

The size and orientation of the TEXT will be determined by the POOL (see Table A.1).

000	○	040		100	@	140	≠
001	□	041	!	101	A	141	≡
002	△	042	"	102	B	142	≤
003	☆	043	#	103	C	143	≥
004	⧖	044	$	104	D	144	±
005	+	045	%	105	E	145	π
006	✳	046	&	106	F	146	Σ
007		047	'	107	G	147	
010		050	(	110	H	150	
011		051	)	111	I	151	
012		052	*	112	J	152	
013		053	+	113	K	153	
014		054	,	114	L	154	
015		055	−	115	M	155	
016		056	.	116	N	156	
017		057	/	117	Ø	157	
020		060	0	120	P	160	
021		061	1	121	Q	161	
022		062	2	122	R	162	
023		063	3	123	S	163	
024		064	4	124	T	164	
025		065	5	125	U	165	
026		066	6	126	V	166	
027		067	7	127	W	167	
030		070	8	130	X	170	
031		071	9	131	Y	171	
032		072	:	132	Z	172	
033		073	;	133	[	173	
034		074	<	134	~	174	
035		075	=	135	]	175	
036		076	>	136	^	176	
037		077	?	137	_	177	

Table A.4 GSYMBL Symbols and Corresponding Octal ICODES

4. Implementation

a. Subroutines

The GS-15 subroutines consist of those intended for users (e.g., GMOVE, GTEXT, etc.) and those written as internal, supportive routines. All user subroutines are written in FORTRAN IV and named according to a standard convention (the first letter of the subroutine name is a "G"). All internal routines are written in assembly language (MACRO-15).

i. User Subroutines: Since the user routines tend to be very short FORTRAN subroutines (typically less than ten statements), their operation can easily be determined by looking at the appropriate source listing. Generally they

(1) verify the validity of parameters,

(2) perform minor calculations,

(3) queue an I/O request node to GLUN (see GQUEUE below), and

(4) return to the caller.

Any error status is returned through the "event variable" (if specified). The POOL--currently a fifty word FORTRAN vector--is declared in the subroutines in a labeled common block named "POOL".

ii. Internal Subroutines: GS-15 uses three internal subroutines that are not visible to the programmer. Two of these are general purpose routines (SETNXT and NUMARG) and the other only useful within GS-15 (GQUEUE).

GQUEUE's sole purpose is to take from three to six of its actual parameters, build them into a standard RSX I/O request node, and queue them to the logical unit number specified in the first POOL element (GLUN). It also zeros all unused I/O request node entries.

SETNXT is actually two routines. The first routine, SETCHR, sets up variables that will later be used by its companion routine NXTCHR, an integer-valued function. SETCHR is invoked as follows:

CALL SETCHR(BUF(1)) .

BUF is a real or double integer array of packed 5/7 ASCII text (i.e., five 7-bit characters per two PDP-15 words). SETCHR simply defines the start of a 5/7 ASCII character string.

The calling sequence for the function NXTCHR is

NXTCHR(0) .

Note that is must be specified with one parameter even though this parameter is ignored. Each call to NXTCHR returns the next ASCII character in the string defined by SETCHR.

The last internal routine used within the GS-15 subroutines is an integer value function, NUMARG. NUMARG returns the number of actual arguments specified in the call to the subroutine in which NUMARG appears. It is called with one parameter--the first formal parameter of the subroutine that calls NUMARG. NUMARG is undefined for subroutines called with zero actual parameters. For example, the following subroutine, TEST, prints out the number of actual parameters in its call:

```
      SUBROUTINE TEST(A,B,C,D,E,F,G,H,I,J,K)
C
      N = NUMARG(A)
      PRINT(16,1)N
1     FORMAT(15)
      RETURN
      END.
```

b. Device Handlers

As mentioned previously, GS-15 currently contains two device handlers (TV and PL). Following is a discussion of the CRT handler, TV. Since these handlers are almost identical, a separate description of PL will not be given.

i. I/O Functions: TV recognizes five standard RSX and five GS-15 I/O function codes. The RSX functions performed by TV include ABORT (01700), ATTACH (02400), DETACH (02500), HINF (03600), and EXIT (77700). Please refer to Chapter Five of the <u>RSX Reference Manual</u> for an explanation of these standard functions.

Since TV required I/O functions not present in any other device handler, new function codes were created. BEGIN (20000) initializes the handler for a new graphics job; END (20100) performs some clean up procedures when a task is finished with the handler. The connection between these function codes and the GBEGIN/GEND subroutines should be clear. Similarly, NEXT (20400) implements the GNEXT subroutine.

MOVE (20200) is one of the most important functions. A MOVE node is queued for each call on GMOVE. The node contains the destination and pen position (UP/DOWN) of a pen movement. A SYMBOL (20300) node is queued by both GSYMBL and GTEXT. Given a

location, character, and character offset, SYMBOL will draw the symbol according to the character size and orientation currently in effect.

ii. Operation: When TV is assigned to a LUN, the handler does some system-dependent initialization (as distinct from BEGIN above) and then waits for a node to be queued. Upon receiving a node, TV jumps to an appropriate section based on the I/O function contained within the node. At the completion of the requested function, the user's event variable is set, the node is removed from the pending request queue, and the next node is processed (if one is present). TV issues a system EXIT directive (thus removing itself from the partition) when the user no longer needs the handler (i.e., the handler is REASSIGNED through MCR).

TV uses three major subroutines to implement the non-standard I/O functions. XYMOVE determines which elementary pen movements are required to move the pen to a new position. This subroutine is based on ACM Algorithm 162 (an ALGOL procedure) and calculates pen movement without the use of division or multiplication.

Each of the ten possible elementary graphics commands (pen up/down, +X, +X+Y, etc.) is assigned a unique octal code. This code is passed to the PLOT subroutine which issues the actual device-dependent IOT (I/O instruction).

SYMBOL is the device handler equivalent of GSYMBL. It is driven by a character table of approximately 400_8 words. Each character in the GS-15 character set (see Table A.4) is defined by a

set of points in a 5 by 7 matrix. SYMBOL interprets the appropriate code word(s), thus producing the required symbol.

APPENDIX B: DIGIT--A DIGITIZING SYSTEM

This program was developed as a utility program to digitize analog data (on-line or from analog tape) and record it on digital magnetic tape for off-line analysis. The present version is a revision which takes advantage of modifications in the analog-to-digital conversion hardware to allow faster sampling rates.

The program uses an overlay structure, defined as follows:

Resident (Root) Section: Executive

Segment D1: Input of data from the operator

I7: High speed data collection

I8: Low speed data collection

D2: Termination of data gathering and closing of tape records

D3: Multiple file; termination of current data collection only, tape not closed

An added capability for this version is a timer which allows data to be taken for a preset interval of time. After DIGIT is requested by the user, Segment D1 is called. It performs initialization and reallocates a partition for use by the actual data collection routine. Data for an identification file is input either from the card reader or from the teletype, and the header file is written on the output tape. When complete, the data acquisition segment (I7 or I8) is run. The I8 (low speed) uses a 5 msec data point interval. The I7 (high speed) uses the A/D clock for timing. This value may be set by varying the

clock. The nominal value is 1 msec. (requires f clock = 2 KHZ). This task inputs data via the A/D converter, and outputs to magnetic tape, using buffered output. Following completion of data collection (as indicated by the user setting ACS 0 or timer runout), segment D2 is run. This segment runs a brief tape verification, printing the header record and scanning for record numbering errors as an indication of data loss. If multiple files are to be written on a single tape, the termination is signified by the user setting ACS 1 instead of ACS 0 (this is indicated for timer runout by setting ACS 17). Segment D3 is then called to perform file heading functions.

USAGE:

1. Verify DIGIT is installed.

2. Verify the device assignments shown below.

```
MCR>DEV
 DK0    1
 TT0    2,3,4,12,13,14,20,21
 RF0    5,10,11,15,17,18
 DT0
 DT1
 MT0    19
 LP0    16
 CD0    22
 PR0
 FP0
 VF0
 AD0
 PL0
 XX0
```

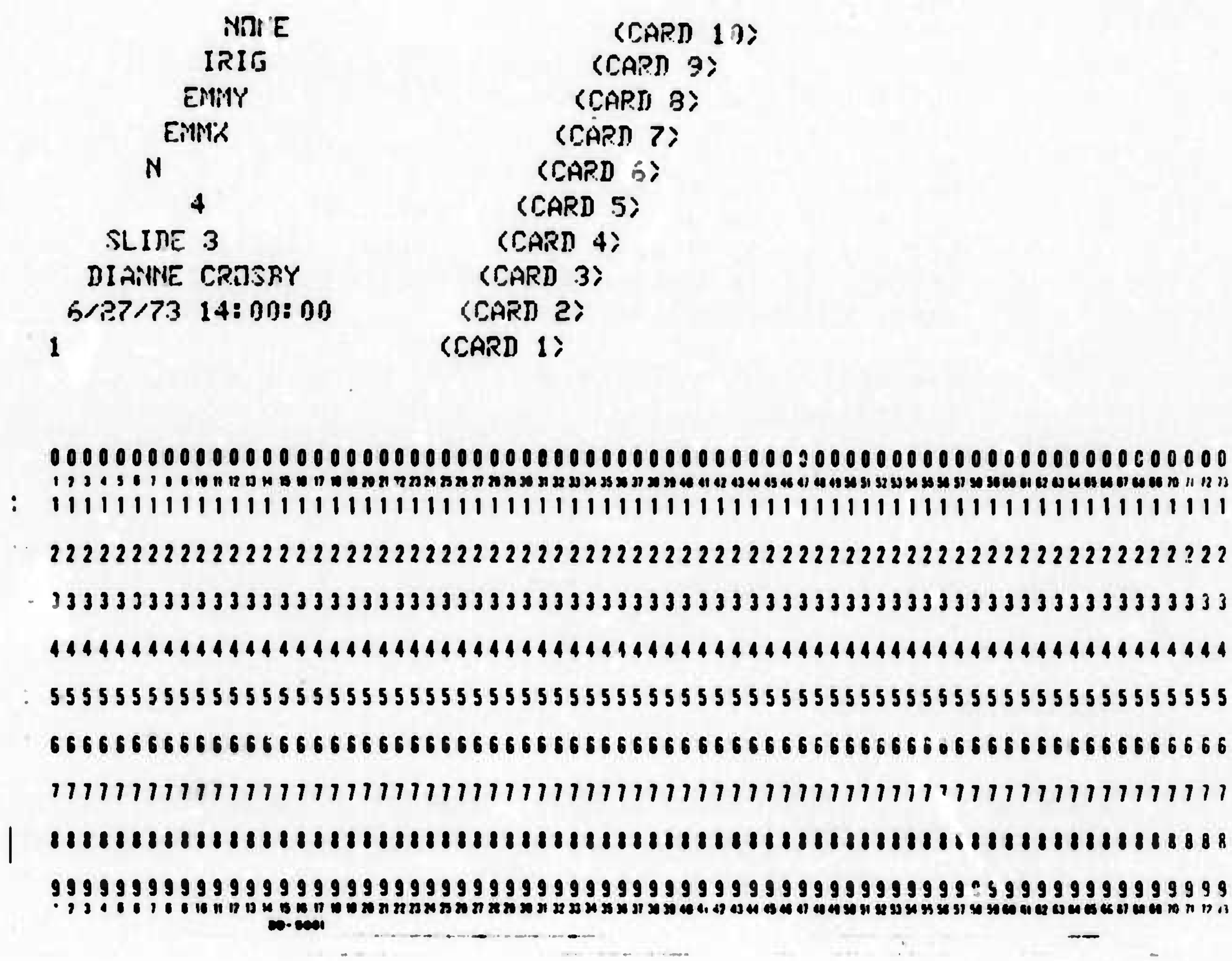
3. The following sequence shows reassign commands often required, and the program operation using the card input of header data.

```
MCR>REA 19 MT DT
MCR>REA 23 NONE PL
MCR>REA 22 CD NONE
MCR>REQ DIGIT]
DATA INPUT FROM CD OR TT?
CD
ENTER RUNTIME IN SECONDS. 0 INDICATES MANUAL
STOP ONLY
20
CR WHEN READY TO BEGIN
SET ACS 0 FOR MANUAL CLOSE, LAST FILE
SET ACS 1 FOR MANUAL CLOSE, CURRENT FILE ONLY
SET ACS 17 TO ENABLE MULTIPLE FILES WHEN USING
TIMED EXECUTION OPTION

***DIGITIZING ACTIVE***
STOP-002222-DIGIT
```

4. The format of the printout is shown below.

```
HEADER FILE CONTENTS

RECORD NO 1          DATE 6/27/73 13:59:00     PROC DATE  7/ 3/73 19:41:41
SUBJ INFO DIANNE CROSBY
DATA INFO NO CHANNELS 4 DIGITIZING RATE 200 HZ   FULL SCALE (12 BITS)=10.0 VOLTS
CHAN IDENTIFICATION:  CH1 EMMX  CH2 EMMY  CH3
COMMENTS  SLIDE 3
RECORD COUNT THIS TAPE =     64
```

5. Card format is as follows:



6. Tape format

The tape format is as follows:

Tape recording method: 9 track, 800 bpi, industry standard NRZI coding.

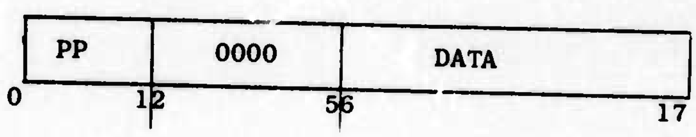
reel size: 2400 feet

number of files: 2

The first file contains 5 records of 81 characters each. The contents are as shown under printout (4) above. The last character on each line is a non-printing character (line feed), coded in EBCDIC format. The file is followed by an end-of-file mark.

The second file consists of a variable number of 256 word records. The number is determined by the length of time the digitizing program is run, and is shown in the tape verification printout (4) as "RECORD COUNT THIS TAPE".

The first four words of each record* are reserved for sequence number and time information. Currently, the time data is not implemented; this will occur after the time code generator interface is operational. The remaining 252 words contain the binary values of A to D converter output, in groups of 4 channels*. The word format is as follows:



PP = parity bits

This format appears when the data is read in using the system directive MTGET.

A system library routine (RIPOFF) is available to convert the

* When using eight channels, the first 8 words are reserved, and the remaining 248 words are in groups of 8 channels.

above format to a FORTRAN compatible single integer word. For use with other operating systems, the data appears as two consecutive 8-bit bytes (or a 16-bit halfword), format as above except the parity bits are omitted.

7. Teletype input of header data

If teletype input is selected instead of cardreader, the following sequence occurs:

```
DATA INPUT FROM CD OR TT?
TT
ENTER RECORD NO
1
ENTER DATE AND TIME RECORDED
6/27/73 13:50:00
ENTER SUBJECT INFORMATION
DIANNE CROSBY
ENTER COMMENTS
SLIDE 3
ENTER NO OF CHANNELS - 4 OR 8
4
HIGH SPEED VERSION ? Y OR N
N
ID FOR CHANNEL  1
EMMX
ID FOR CHANNEL  2
EMMY
ID FOR CHANNEL  3
IRIG
ID FOR CHANNEL  4
NONE
```

Each input is terminted by a CR. All D-numbers will have only the number entered (i.e., do not enter D-9; enter 9).

APPENDIX C: MISCELLANEOUS UTILITY SOFTWARE

1. CORE/DISK DUMP

An RSX user may examine the contents of the disk or memory through use of the MCR function, OPEN. OPEN allows the inspection of only one word per invocation; however, it is unsuited for displaying large, contiguous blocks of core and disk. For this reason, an additional MCR function, DMP, was written. The syntax definition of the DMP command is as follows:

DMP␣[D] from [␣to]

where from = starting core/disk address

to = ending core/disk address

from $\leq$ to.

Both the starting and ending addresses must be octal constants. The optional parameter, "D", is specified when a disk dump rather than a core dump is required. The dump listing will be written on logical unit LUN 16 which is normally assigned to the line printer. DMP first extracts the starting and ending addresses from the command line, and then initializes itself by filling the output buffer with blanks. As each word is received from core or disk, it is placed into the output buffer. When the buffer is full (eight words), it is printed on LUN 16. This procedure continues until the requested area is dumped. The flowchart for DMP is shown in Figure C.1.

2. DECtape Dump

A program (DTDUMP) is available to DUMP in octal, selected blocks of a DECtape assigned to LUN 19. Since DTDUMP is a task rather than an

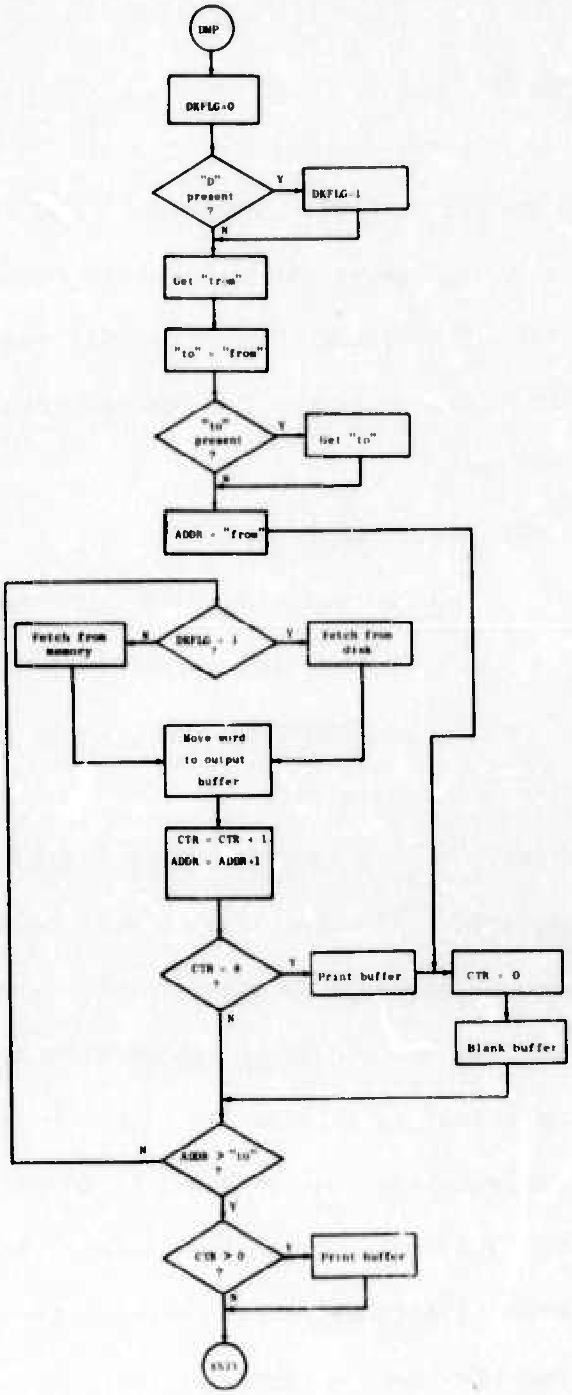


Figure C.2 Flowchart for CORE/DISK DUMP Program

MCR function, its execution is intiated through the REQUEST function of RSX. The user is asked to specify the beginning and ending blocks to be dumped after which a listing is produced on LUN 16 (line printer). The program re-cycles until a starting block of zero is given in response to DTDUMP's question.

3. Magnetic Tape Dump

This is a system utility program written to print out the data on 9-track, 800 bpi industry compatible magnetic tape, in an octal format. It is a task, called by the system REQUEST function. The program uses the console teletype for user input of the expected record size and the number of records. Each record is read in, the format converted, and dumped to the printer. Printing suppression is used for all zero lines. The program will run even if the incorrect record size is specified, with the extraneous values suppressed. Termination occurs on end of file, or completion of the required number of records. No rewind is built in to the program to allow restarting the dump after a known number of records, or after some significant event on the tape determined by a preceding program. A brief flowchart is shown in Figure C.2.

4. IAS/ISWTCH

The PDP-15 has a group of eighteen "data switches" or "sense switches" (numbered 0, 1,...,17) mounted on the operator's console. Since the positions of these switches may be dynamically determined through the execution of a hardware instruction (OAS), programs may use these switches for low volume data input. The standard DEC soft-

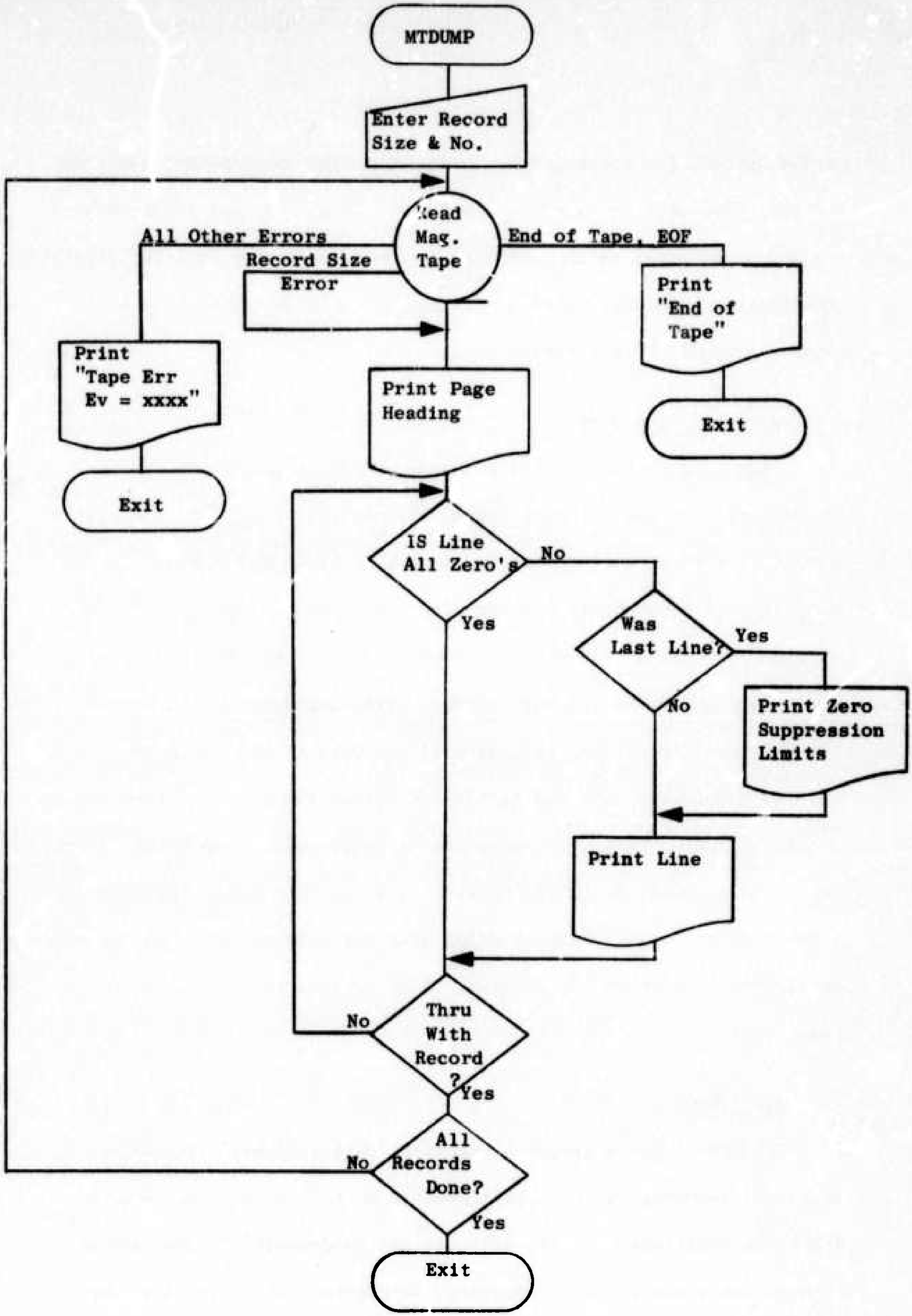


Figure C.2 Flowchart for MTDUMP Program

ware allows only assembly language programmers to determine the data switch settings. In order to extend this capability for FORTRAN programmers, two FORTRAN callable routines have been created: LAS and ISWTCH.

The eighteen data switches may be thought of as a single eighteen-bit binary integer since each switch must be in only one of two possible positions. When a user wants to ascertain the settings of all or most of the switches, one may include the INTEGER FUNCTION LAS in his program. When calling LAS, an argument must be specified. For example, the following short FORTRAN program will print the current status of the data switches as a six-digit octal integer:

```
      N = LAS(0)
      PRINT (13,1) N
1     FORMAT (1X,Ø6)
      END
```

Frequently, a program contains a number of optional features that the user must selectively request. A simple way to tell this type of program what options are needed is to assign one data switch per option. In order to allow FORTRAN programs to test single data switches, a LOGICAL FUNCTION ISWTCH was written. This routine is referenced as follows:

ISWTCH(I) where I = INTEGER variable or constant ($0 \leq I \leq 17$)

ISWTCH(I) returns a "TRUE" value if and only if data switch "I" is on. A FORTRAN statement that would pause whenever both data switches #0 and #6 were on is coded as

IF (ISWTCH(0) .AND. ISWTCH(6)) PAUSE.

A flowchart depicting ISWTCH's operation is given in Figure C.3.

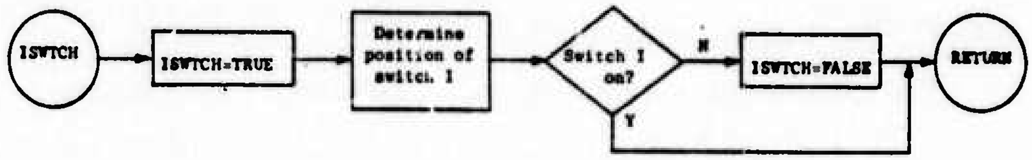


Figure C.3 Flowchart for ISWTCH Program

5. RIPOFF

This subroutine was written to convert data recorded on magnetic tape, or analog-to-digital converted data to a format consistent with the internal formats for data values. Certain magnetic tape input/output routines pass the data to the calling program without removing the parity bits. Also, A/D converter data is 12-bits wide. Thus, negative values are not properly represented in an 18-bit word. This subroutine corrects for both problems. It is currently in use as a FORTRAN callable library routine.

6. REA

The RSX operating system utilizes core partitions outside of the monitor for input-output device handlers. The use of these partitions is not allowed by other tasks if the device is on line and in use by the system. This subroutine was developed to dynamically reallocate the core space used by these handlers by reassigning the device off line. This allows the use of a peripheral device (such as a line printer) to be deferred during real time operations (such as digitizing),

then returned for use during a following off-line operation, under program control. It is a FORTRAN callable routine, currently used in the DIGIT program.

7. GT132

One other special purpose subroutine is included in the disk-resident user library. The routine was created because of an early need to read digitized EEG magnetic tapes. Normally these tapes would have been read by the standard FORTRAN input statements. However, at the time these tapes were needed, a FORTRAN program could not read them because of errors in the hardware and/or software.

Therefore, a subroutine, GT132, was written which would read one record (132 characters long) from a digitized EEG tape and pass this record to the user. The calling sequence for GT132 is

```
CALL GT132(BUF,EOF)
      where BUF = 27 element DOUBLE INTEGER array
            EOF = FORTRAN statement number .
```

Each call to GT132 places a tape record into BUF unless an end-of-file is encountered. In the latter case, GT132 returns control to the statement specified in EOF instead of executing a normal RETURN statement.

As shown in the flowchart in Figure C.4, GT132 translates the EBCDIC characters on tape into the ASCII character equivalents required by RSX after first reading the tape in binary. DEC just recently corrected the problem so that we are now able to read the tapes with the FORTRAN READ statement.

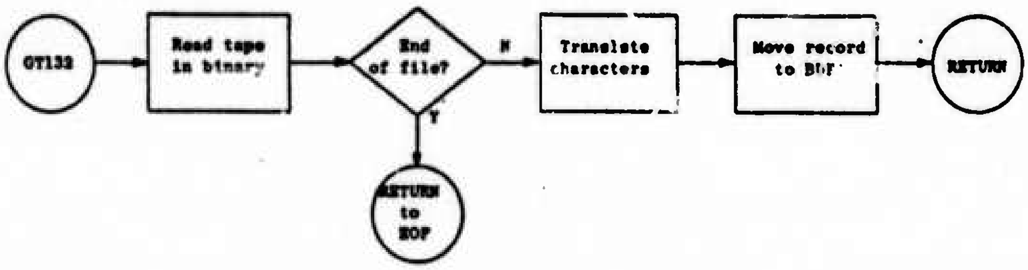


Figure C.4 Flowchart for GT132 Program

8. EMMXY

The current version of EMMXY plots X vs. Y values from standard digitized tapes (see Appendix B) of eye-movement data. The calibration record for the data is assumed to be on disk.

The user is queried as to the channel for X data (1 to 4), the channel for Y data (1 to 4), and the compression (compression is defined as the number of points to skip between plotting points; it is a method of editing out redundant data points). The program goes into a wait mode whenever data switch 0 is set to a 1 state, and exits on end of tape or whenever data switch 1 is set to a 1 state.

This is a baseline program. Several version of this program are used by various individuals on the project.

Usage: (Verify LUN-23 = PLO)

MCR > REQ EMMXY

ENTER COMPRESSION

5

ENTER CHANNEL NO - X

1

ENTER CHANNEL NO - Y

2

9. CALDT

This program allows the current calibration record (disk file "CALRC SRC") to be written on DECtape unit 0 with a user selected file name (extension "SRC"). The default name is "CALRC".

Usage: (Verify LUN-19 = DT0)

MCR > REQ CALDT

ENTER FILE NAME

xxxxx — User inputs file name, 5 characters maximum. To select default, CR only.

STOP - 2222 - CALDT

10. CALTD

This program retrieves a calibration record previously stored on DECtape and prints the values. It may then be stored on disk as the current calibration record for analysis programs.

Usage: (Verify LUN-19 = DT0)

MCR > REQ CALTD

ENTER FILE NAME

xxxxx

FILE TO DISK? Y OR N

Y

STOP - 2222 - CALTD

11. CALPLT

This program allows plotting of the calibration point matrix. The scaled values of calibration data for each point may then be overlayed from the current calibration record. The pen may be positioned in a position (0,0) to allow following data to be directly overlayed, or the pen may be positioned so that subsequent data will not overlay the current record.

Usage: (Verify LUN-23 = PL and LUN-22 ≠ CD)

MCR > REQ CALPLT   The matrix is plotted

CAL DATA ALSO? Y OR N

Y

POSITION PEN FOR OVERLAY? Y OR N

Y

STOP - 2222 - CALPLT

12. CALCHK

This program accesses the current calibration record on disk and prints the values. No modification is made to the record.

Usage: (Verify LUN-16 = LP)

MCR > REQ CALCHK

STOP - 2222 - CALCHK

APPENDIX D: APPLICATIONS SOFTWARE

1. Eye-Movement Data Acquisition and Analysis

a. Introduction

This section describes the programs designed for use in acquisition and analysis of the eye-movement data. It is desirable to standardize the operation of programs, in so far as practical, to obtain the maximum usage from any given software "investment". Thus a similar approach will be taken for EEG data.

b. Data Flow

The software flow of data under this processing scheme is as follows:

(1) A calibration program is run to complete an overall system calibration. In the case of eye-movement data, this involves positioning the subject's eyes at predetermined locations. For "dimensionless" measurements, such as EEG, a voltage check will be made.

(2) All significant data are written to the disk at the time of calibration. This allows subsequent analysis programs to access the data. It also protects against system crashes during following real-time programs (if the data were in core it would be lost).

(3) Measurements are made using the current version of the DIGIT program. The data may be entered directly, or via analog

magnetic tape. A standard format is used for the digitized data tape, including an identification file and a data file.

(4) Analysis programs may then be run. Examples are plotting and computational programs.

(5) Utility programs are used for transfer of disk-stored data.

(6) Manual procedures are required for data logging (i.e., tape and record identification numbers).

c. Program Description

This section contains a brief explanation of the calibration program (EMMCL5). It is basically user oriented and is the current release of the eye-movement monitor calibration program. The program contains a prestored sequence for the 9-point calibration chart. The subject is directed to fixate on the points as directed on the console teletype. Following all data gathering, the full scale, zero shift and cross coupling terms are calculated. The plot capability involves calculating scale factors for plotting and writing these values to the disk. If immediate plotting is selected, the Cal matrix and corrected values are plotted (otherwise this may be done at any time up to the next Cal by utility program CALPLT). An additional capability is to input data without using the prestored sequence. In this case, only scaling to volts is performed - no attempt is made to calculate the normal Cal terms.

Usage: Via the GS-15 graphics system, the plot output may be on a CRT or the CalComp plotter. This program uses an overlay structure to perform its functions within core constraints. The segments are as follows:

Resident (Root) Section: Calling or executive only.

Segment A: Input of data from the operator; input of eye-movement measurements.

Segment B: Manipulation of the data and scaling.

Segment C: Calculation and plotting; disk storage of data.

Verify LUN 23 is assigned to PLO.

Verify correct data and time via DATE function.

MCR > REQ EMMCL5

(1) ENTER SUBJECT ID

xxxxxxxxxx	10 characters maximum, followed by CR.

(2) SET UP EYE MOVEMENT MONITOR AND CALIBRATE MANUALLY, NL WHEN READY — Issue CR when ready

(3) PRINT RAW DATA? Y OR N

N	Followed by CR. This feature is not required for normal runs.

(4) PLOT DATA? Y OR N

Y	This selects immediate plotting of the calibration data.

(5) PRESET CAL SEQUENCE? Y OR N

Y	This selects the normal calibration sequence for the 9-point calibration array.

(6) DIRECT SUBJECT TO POSITION PRINTED. NL WHEN READY.

5

Note: The data is taken for each point <u>after</u> the CR is entered.

8

5

.

.

.

.

.

9

5

At the end of the sequence, the summary data is printed. See below for format. The following message is printed if the plot option is chosen.

(7) POSITION PEN FOR OVERLAY? Y OR N

Y

Followed by CR. In this case, the pen is returned to (0,0) so that a following data analysis program may overlay the plot on the same coordinate system. If N is chosen, the system goes to a location to allow following plots to be made without inteference.

(8) STOP - 2222 - EMMCL5 Signifies end of program.

If the (5) PRESET CAL SEQUENCE is not chosen, the following message is printed.

(9) ENTER POSITION, FOLLOWED BY CR. MAX OF 23 POINTS

Note: as above, the data is taken after the CR. The summary data printout contains only the scaled data values. If this option is chosen, the plot sequence is not entered, and no data is written to the disk.

(10) STOP - 1111 - EMMCL5 Signifies end of program.

2. SEER

We have recently begun work on a program that will enable us to do on-line analysis of visual methods of object recognition. SEER, unlike our earlier eye movement experiments, is designed to use the Tektronix 611 CRT for image production and viewing. Although the design is still being developed, certain features have been decided upon in the SEER system.

a. Operation

The most basic entity in SEER is the "picture". A picture corresponds to a set of data collected together and given a symbolic name. The data composing a picture consists of an ordered set of numbers which, when processed by the GS-15 graphics system, produces an image on the surface of the CRT. These images are then viewed by subjects while SEER monitors their eye movements, thus enabling us to quickly determine scanpaths, etc.

We will build up a "library" of such pictures on DECtape - one DECtape file being equivalent to a picture. To initiate an experiment, the picture library will first be mounted on a DECtape unit. The user must then specify which pictures in the library will be used. These pictures are read from DECtape and collected together in the picture "album" - a random access disk file. Subsequent SEER commands can then refer to any picture in the album and/or make additions to or deletions from the album.

b. Modularity

SEER contains a two-level command hierarchy. Major command functions allow picture composition (CREATE), picture viewing (DRAW), and album updating (ALBUM). Each major command sub-system has its own minor commands as described in the next section.

Because of the modular design implicit in a major/minor command structure, new features may be easily added as our experimental sophistication increases. Furthermore, since almost all of the software will be written in FORTRAN IV, the system could be transferred to another computer system.

c. Major Sub-Systems

i. ALBUM: Each picture in the album has a six character symbolic name (e.g., FACE, SQUARS, etc.). Pictures may be removed from (DELETE) or added to (INSERT) the album. A listing of the album (i.e., which pictures are present) may be obtained through use of the CONTENTS minor command. In addition, pictures in the album may be stored on magnetic tape by using the SAVE command.

ii. CREATE: The purpose of this sub-system is to create new pictures. Minor commands exist to "paint" geometric objects such as lines, rectangles, and circles. Furthermore, the TABLET command allows complex, non-linear pictures to be entered through a writing table. Other commands NAME newly created pictures and permit additions to pictures already in the album (APPEND).

iii. DRAW: In this sub-system pictures in the album may be drawn

on the CRT. A picture may be simply displayed (PREVIEW) or displayed while eye-movement data is being taken on the subject (VIEW). Also, the ERASE minor command will clear the CRT screen.

LIST OF PUBLICATIONS

(a) D. C. Lai and R. L. Lux, "Application of Frequency Discrimination Technique to the Analysis of EEG Signals", Proceedings of the National Electronics Conference, vol. 27, pp. 80-85, Oct. 1972.

(b) R. V. Floyd, D. C. Lai, and J. E. Anliker, "A Model for the Photically Stimulated Electroencephalographic Signals", Presented at the 12th Annual San Diego Biomedical Symposium (31 January - 2 February 1973). To be published in the Proceedings.

(c) K. Jacker, "The Graphics Software DEC Forgot to Include", Presented at the DECUS Spring Symposium (2 May - 5 May 1973). To be published in the Proceedings.

(d) M. Ein-Gal and D. C. Lai, "Real-Time EEG Analysis and Monitoring Using In-Phase and Quadrature Components", To be presented at the 26th Annual Conference on Engineering in Medicine and Biology (30 September - 4 October 1973).

(e) M. Ein-Gal and D. C. Lai, "Error-Free EEG Signal Representation", submitted for presentation at the 1973 International Systems Man and Cybernetics Conference (5 November - 7 November 1973) and for publication.